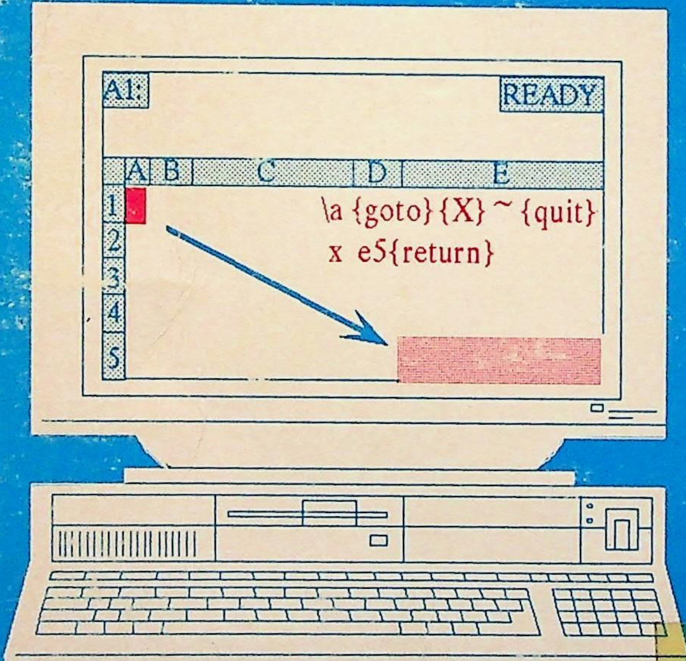
רן אברהמי

תורת המקרו בלוטוס ותואמיו

חלק א

יסודות

ספר ללימוד עצמי



BUG
המרכז לספרות, תוכנה ומחשבים

גישה חדשה ללימוד עצמי של יסודות המקרו על בסיס נסיונו העשיר של המחבר בהוראה ובכתיבת יישומים

על ידי מחבר הספר מקרו למתקדמים בלוטוס

רן אברהמי

תורת המקרו בלוטוס ותואמיו

חלק א: יסודות

ספר ללימוד עצמי

ספרים אחרים על ידי רן אברהמי:

מקרו למתקדמים בלוטוס Meta Macro ו Intra Macro בהוצאת הוד עמי

רן אברהמי

תורת המקרו בלוטוס ותואמיו

חלק א: יסודות

ספר ללימוד עצמי

ספרים אחרים על ידי רן אברהמי:

מקרו למתקדמים בלוטוס Meta Macro ו Intra Macro בהוצאת הוד עמי

תורת המקרו בלוטוס ותואמיו חלק א: יסודות

כתב וערך: **רן אברהמי**
יועץ מיחשוב
ת.ד. 4305 ירושלים 91042 טלפון 02-631768

ספר זה נכתב (טקסט, משוואות וגראפיקה), נערך וסודר לדפוס ע"י תוכנת לוטוס 1-2-3 גירסה 2.3 עברית.

הודפס בדפוס המכפיל בע"מ - ירושלים

Macro Theory In Lotus And Compatibels   Part 1: Basics

by Ran Avrahami

Computer Consultant
P.O.B 4305 Jerusalem 91042 Israel Tel. 02 - 631768

הדפסה לפני צילום לדפוס בוצעה במדפסת לייזר באדיבות חברת Computerland בע"מ - מרכז ירושלים

מוקדש באהבה לילדי

גלי

דנה

טל

תואמות ואוטומציה

מאחרי כל קניה של מחשב מקננת הציפיה שבעזרתו אפשר ליעל תפוקה, כלומר להקטין את משך הזמן לביצוע מטלות שונות ולהניע מספר רב יותר של מטלות בצנרת. מילת המפתח היא - אוטומציה. הראש ממש תופח בציפיה לגדולות ולנצורות שהאוטומציה תבצע עבורנו. ולכן, בדיעבד, ליחידה עיסקית השתלם לשלב מחשב בפעילותה רק אם עובדיה השכילו ליעל תפוקה על ידי אוטומציה. וכשיש מחשב צריך תוכנה וכשיש תוכנה צריך ללמוד להפעיל אותה.

שימוש בגיליון אלקטרוני, יהיה אשר יהיה, דורש השקעת זמן כדי ללמוד להפעילו. ככל שרמת הידע באוטומציה על ידי הגיליון האלקטרוני עולה כך גם עולה יכולתו של המשתמש ליעל ביצוע מגוון רחב של מטלות - שלו ושל אחרים באירגון. בגיליון אלקטרוני אפשר להגיע לביצועים מדהימים בהשקעת זמן מינימלית. וחשוב לא פחות שליכולת כזאת, כשמדובר בגיליון אלקטרוני, יכולים להגיע עובדים מתוך האירגון מבלי שיהיה צורך להתקשר עם מתכנתים ויועצי מיחשוב מבחוץ - ששם באמת מתחילות העלויות להצטבר. אפשר לאבחן ארבע רמות כשמתקדמים במסלול לרכישת ידע לאוטומציה בעזרת הגיליון האלקטרוני.

1. הכרת הגיליון
2. תורת המקרו
   1. יסודות
   2. שליטה
   3. מערכות

כאן נתקלים לראשונה בבעית התואמות. חלק ניכר מעלות ההתקדמות לקראת אוטומציה באירגון על ידי מחשב הוא השקעת האירגון בעובדים - לצורך הלימודים צריכים העובד והמעביד להשקיע זמן, מזומן ורצון - ביחוד זה האחרון. ואחרי כל ההשקעה המצטברת הזאת מופיעה גירסה משופרת של אותו יצרן או מתחריו - ומה? צריך להתחיל הכל מחדש? ובכן, המצב הנוכחי בשוק תוכנות הגיליון האלקטרוני מאפשר פיתרון יעיל ואלגנטי לבעית התואמות.

עד כה, הגיליון המוביל בעולם הן במספר המשתמשים והן ביצירת סטנדרד הפעלה הוא לוטוס 1-2-3 . כל זמן שמצב זה קיים אפשר לאמר כי

כל שתלמד בלוטוס יכשירך לתפעל באותה רמה כל מוצר הטוען לתואמות מלאה איתו.

כאן מונח הפיתרון לבעית תואמות בין מוצרים. ומה בקשר לתואמות בין גירסאות לוטוס שונות? בהתקדמות לקראת יכולות אוטומציה גבוהות עברו גירסאות הלוטוס מספר שלבים. קפיצה משמעותית ביותר התרחשה בגירסה 2.01 שהעלתה את יכולות האוטומציה בגיליון לרמה מדהימה. לאחר מכן כל גירסה חדשה כוללת שיפורים חשובים - חלקם מהיזון חוזר ממשתמשים מתוחכמים וחלקם מפיתוחים מקוריים של יחידת הפיתוח של לוטוס. כל גירסה חדשה יכולה להפעיל יישומים שהוקמו בקודמות. לגירסה 2.2 מקום טוב באמצע מכיון שבשינויים קלים מאוד אפשר להפעיל יישומים שפותחו בה גם בגירסה 2.01.

עד כאן עסקנו בתואמות בין מוצרים בהקשר ליישום שנכתב במוצר אחד והפעלתו ע"י מוצר אחר. ומה באשר לידע שנרכש במוצר אחד?

יסוד היסודות בהכשרת תלמידים באוטומציה בעזרת גיליון אלקטרוני הוא פיתוח יכולת חשיבה יוצרת ורכישת כלים ונוהלים לפתרון בעיות באופן עצמאי. משרכש התלמיד את הידע הזה - יחד עם הטכניקה של שפת המקרו בגיליון - יוכל לפתור בעיות מזדמנות בכל רמת מוצר ובכל סוג מוצר שיזדמן לו. וככל שתעלה רמת הידע שלו באוטומציה כך גם תגבר יכולתו זו.

שמו של ספר זה מעיד על יעודו. הוא נועד למי שמכירים כבר את הגיליון האלקטרוני ורוצים ללמוד את יסודות המקרו. נעשה מאמץ רב לכתוב את הספר בצורה שתאפשר לימוד עצמי - אם הצלחתי בכך רק אתם תגידו. כל התרגילים מתבססים על גירסת לוטוס 2.2

תוכן העניינים הקצר

תוכן העניינים

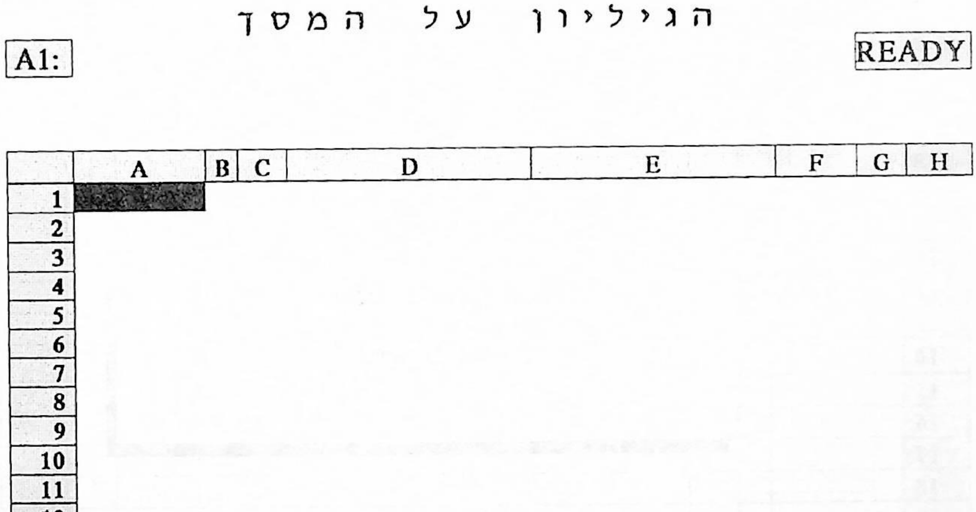
פרק א: הולכת סמן

1.1 המפה והשטח

יש בנו דחף פנימי המניע אותנו כל הזמן לרצות להשפיע על כל הסובב אותנו, לחולל בו שינויים ולהתאימו לתפיסת העולם שלנו. כדי שאפשר יהיה לחולל שינויים כאלה חייבים להתקיים מספר תנאים - מקום עבורנו ומקום עבור האוביקט שרוצים לשנות (גם כשהאוביקט הוא אנחנו); כתובת לכל אחד משנינו כדי לדעת את מיקומו של האוביקט יחסית למקומנו (לכך דרושה מפת האיזור בו אנו ממוקמים); ולבסוף דרוש נוהל תיקשורת, כלומר שפה: שפת גוף, שפת דיבור, שפת כתיבה, שפת מחשב ...

הבה נבחן את הגיליון האלקטרוני לאור הדברים הללו.

איור 1.01
הגיליון על המסך



כשנטען הגיליון רואים על המסך את התמונה המובאת באיור 1.01. המסך הוא מקום, סביבה נראית לעין שבתוכה יש או יהיו אוביקטים

שאותם נרצה לשנות. בעת טעינת הגיליון ממוסגר המסך ע"י שני סרגלים - האחד צמוד לשפתו השמאלית של המסך ובו מספרים רצים בסדר עולה והשני ניצב אל הראשון ומכיל אותיות באנגלית לפי סדר הא"ב משמאל לימין, ושלוש שורות מפרידות בינו לבין שפת המסך שמעליו. במקום מפגש שני הסרגלים בפינה השמאלית העליונה רואים מלבן בוהק. איור 1.02 עוזר להבהיר את משמעות הסרגלים הללו.

איור 1.02
הגיליון כמפה



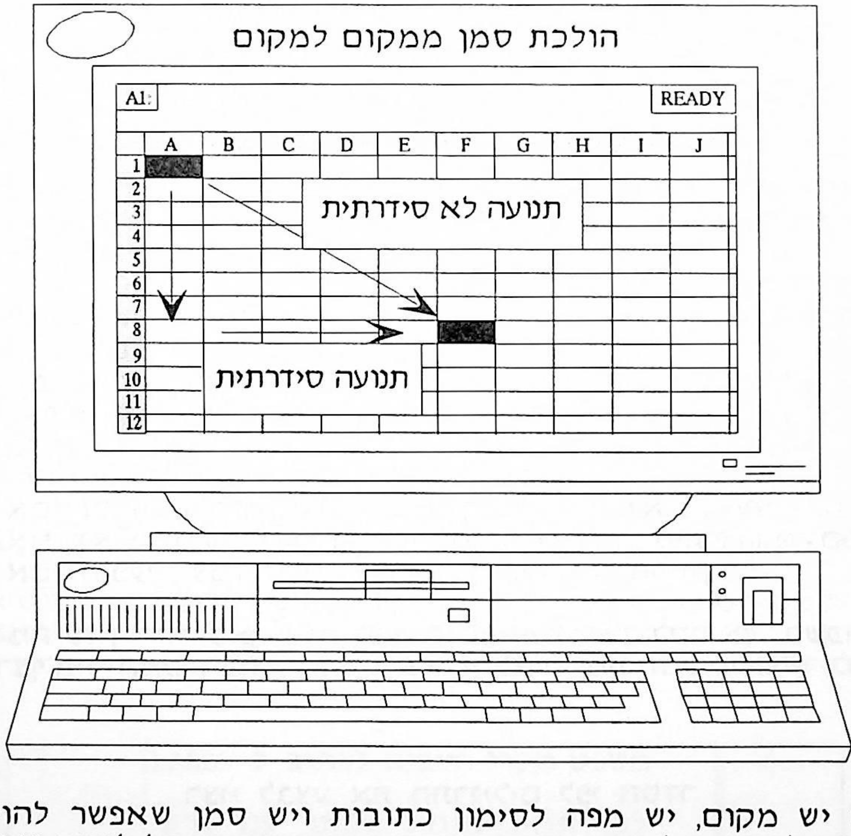
המסך הוא המקום או השטח שבו ימוקמו דברים ועליו יראו השינויים שיתרחשו והמסך הוא המקום בו תתרחש תיקשורת דו-צדדית בינינו לבין המחשב. אלא שעל מסך במערומיו אי אפשר לציין מקומות באין מסגרת התייחסות. סרגלי הגיליון האלקטרוני הם מערכת התייחסות כזאת, מערכת קואורדינטות בדיוק כמו בכל מפה. בין הסרגלים המפגש של

קו דימיוני לאורך עמודה עם קו דמיוני שני הניצב אליו ונמשך לאורך שורה הוא כתובת מורכבת מהאות בסרגל העליון בראש העמודה הזאת וממספר השורה בסרגל שמשמאל, למשל B7.

בין הסרגלים מעמיד לרשותנו הגיליון מתקן להצביע, כמו עפרון, על מקומות במסך. לסימון נקודה במפה מסתייעים במד קואורדינטות כדי לקבל את הכתובת המדויקת ואילו בגיליון הכתובת מחושבת אוטומטית וקוראים אותה בזמן אמת בפינה השמאלית העליונה של המסך מעל לסרגל העליון. הולכת הסמן משנה כתובת זו בהתאמה.

מפה זו מיוחדת במינה מכיון שלא כמפות אחרות משולבת היא בשטח וכל פעולה המציינת כתובת (ואיזו לא?) במפה גורמת מיד לשינוי בשטח (במסך).

איור 1.03
בגיליון המפה היא השטח



והנה יש מקום, יש מפה לסימון כתובות ויש סמן שאפשר להוליכו ממקום למקום על פני המפה וחסרה רק שפה כדי שנוכל לגרום שינויים

על גבי המסך. לפני שנדון בשפת המקרו נבחן מספר מרכיבים של שפה.

תפוקתו של משפט היא המשמעות שמייחס השומע למה שנאמר לו. בשפת אנוש שומעים שונים עשויים לייחס משמעות שונה לאותו משפט ושומע אחד עשוי לייחס למשפט אחד משמעויות שונות בזמנים שונים. גם בשפת הכתב יש ריבוי משמעויות, ריבוי המעניק לשפה עומק ויופי יחד עם אי בהירות עקב ריבוי תפוקות למשפט אחד. כאן יש שוני בין שפה רגילה לבין שפת תיקשורת עם המחשב. בשיחה עם המחשב יש לכל מלה משפט או פסוק משמעות אחת חד ערכית שגורמת למחשב להניב תפוקה זהה לאותו משפט בכל עת.

1.2 מקרו לתנועה בגיליון

1.2.1.1 סמן בפועל - תנועה סידרתית

משפט מקרו הוא מרכיב בשפת התיקשורת עם המחשב וכמו בכל שפה הוא מוכלל במסגרת שמפרידה אותו ממשפטים אחרים:

1. התחלת המשפט,
2. שרשרת סימנים ליצירת תוכן,
3. סוף המשפט.

משפט הוא רצף תווים שתחילתו בראשון וסופו באחרון. התו האחרון הוא הנקודה, לאמור - סוף המשפט. נוהל קל ונוח לעין אבל לא מובן למחשב. "הדיבור" אל המחשב מתרחש בערך כך:

1. יש בגיליון תא וכתובתו, נאמר, a =
2. גש לשם
3. תמצא שם רצף תווים
4. רצף זה הוא משפט, או חלק ממנו, שאומר מה לעשות
5. בצע
6. כשתגיע לקבוצת התווים {quit} - הפסק
7. אם אין {quit} בתא הנוכחי המשך לקרוא בתא מתחתיו
8. אם הוא (מתחת לקודמו באותה עמודה) ריק או הגעת ל {quit} - הפסק
9. אם הופעל צמד המקשים Cntrl Break - הפסק

דבר מה עדין חסר כאן (נסה למצוא לפני שתמשיך לקרוא). כשכתבנו את רצף תווי המקרו בתא כל שהוא עדין לא הקמנו נוהל שיאפשר להגיד

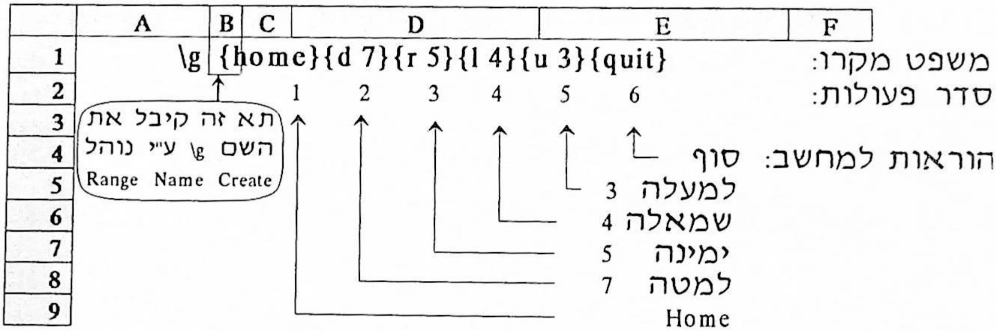
רצוי לבצע את התרגילים לפי הסדר

למחשב - שים לב, לך לשם, פעל. לשם כך נבצע מספר פעולות מכינות:

1. Range Name Create אך ורק בתא בו מתחיל משפט המקרו (b1 בתרגיל)
2. שם התחום יהיה רק בן שני תווים: אות אחת מהא"ב האנגלי והתו \ (אלכסון הפוך - back slash). בתרגיל הנוכחי - g\
3. נכתוב g\' בתא שמשמאל למקרו (a1 בתרגיל) לדוקומנטציה.

עכשיו, משנקיש על צמד המקשים Alt g יבין המחשב ויבצע את הדברים הרשומים בסדר הפעולות של תרגיל 1.01.

תרגיל 1.01
תחליף למקשי החיצים



עתה נפנה לתרגיל 1.01. אפשר לשלב במשפט מקרו קבוצות תווים בנות משמעויות שונות:

1. פקודת מקרו
2. פונקציה
3. פקודת מקרו בתוך פונקציה
4. מספר
5. טקסט עברי\אנגלי
6. פקודה מתפריט הלוטוס

פקודת מקרו מורכבת ממלת הפעלה ולעיתים קרובות גם מארגומנט אחד או יותר. מלת ההפעלה והארגומנטים שלה מוכללים בתוך סוגריים מסולסלים {...} כדי להבדיל בין הפקודות השונות. הבה נבחן את

משפט מקרו חייבים לכתוב כ LABEL לכן תמיד נתחיל אותו עם גרש

הפקודות בתרגיל 1.1 לפי סדר הפעולות של המשפט.

1. פקודת המקרו {home} מכילה מלת הפעלה ללא ארגומנטים והיא מורה למחשב לבצע כאילו הקשנו ידנית על המקש Home.

2. פקודת המקרו {d 7} מכילה מלת הפעלה וארגומנט אחד וביניהם מבדיל תו רווח אחד. המלה d מורה למחשב לבצע פעולה זהה לזו שהיתה מתרחשת אילו הקשנו חץ למטה. הארגומנט 7 מורה לבצע את הפעולה 7 פעמים. אם נכתוב {d} ללא ארגומנט ברירת המחדל היא {d 1}.

3. פקודת המקרו {r 5} מורה למחשב לפעול כאילו הקשנו ידנית חץ ימינה 5 פעמים.

4. פקודת המקרו {l 4} מורה למחשב לפעול כאילו הקשנו ידנית חץ שמאלה 4 פעמים.

5. לפקודת המקרו {u 3} מבצע המחשב 3 הקשות על החץ למעלה.

6. {quit} מודיע למחשב - סוף פסוק למשפט מקרו זה.

הפעל את הגיליון האלקטרוני שלך, הקם בו את המקרו הזה והפעל אותו על ידי הקשה על צמד המקשים Alt g.

ניתן לראות את פעולת המקרו לאט לאט לפי סדר הפעולות. לשם כך הקש קודם על צמד המקשים Alt F2 ואח"כ על הצמד Alt g. משפט המקרו שהופעל יופיע בתחתית הגיליון מצד שמאל והקשה על כל מקש תגרום ש ...

המקרו יתבצע צעד צעד כל פעם תו אחד בלבד כשהסמן על המקרו בתחתית הגיליון משמאל נע כל פעם ימינה ובתוך הגיליון רואים את הביצוע בשלבים (Step Mode). לצאת מ Step Mode הקש Alt F2 שנית.

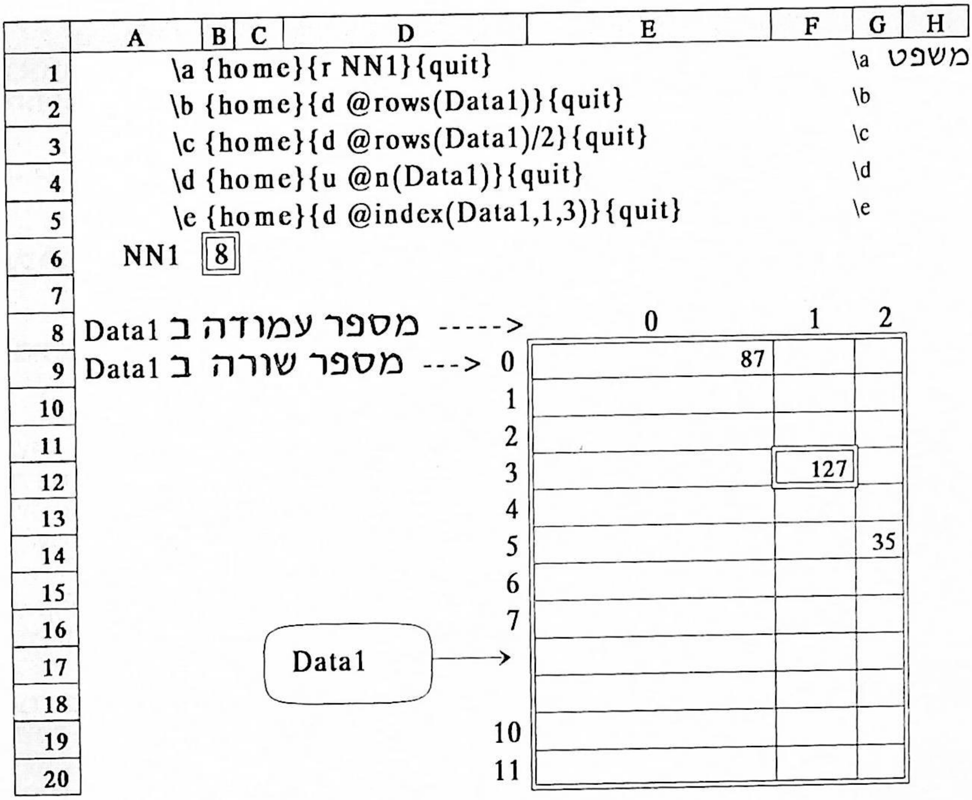
כתוב משפט מקרו שיוליך את הסמן בצורה של מרובע.

כתוב משפט מקרו שיוליך את הסמן בצורה של משולש.

פקודת מקרו

{Command Argument1,Argument2,...ArgumentN}

תרגיל 1.02
ארגומנטים בתחליף למקשי החיצים



בתרגיל 1.01 פעולה 3 כתבנו: {r 5}. הארגומנט 5 בפקודה זו הוא מספר קבוע שאפשר לשנותו ידנית בלבד בכתיבה במקרו בטרם הופעל. יש מטלות רבות הדורשות שהמקרו "ידע" מה מתרחש בגיליון - אחרי שהופעל. למשל אפשר לכתוב מקרו שישנה הגדרה של תחום ואח"כ לתת פקודה שתציב את הסמן שורה אחת מתחת לתחום כדי להוסיף נתונים. לכך דרוש נוהל שיאפשר להחדיר לפקודת המקרו שינויים במספר

בגירסאות הראשונות מלת ההפעלה היא {down} ולא {d} {right} ולא {r} {left} ולא {l} {up} ולא {u}

מקרו כותבים עד 240 תווים בתא. אח"כ אפשר להמשיך לכתוב את המקרו בתא שמתחתיו באותה עמודה

הפעמים להולכת הסמן באופן אוטומטי.

תרגיל 1.02 מדגים טכניקות להשתלת משתנה דינמי כארגומנט בפקודת מקרו. הבה נראה.

נאמר שבהכנה ידנית מוקדמת ייצרנו תחום בשם NN1 בעזרת פקודת התפריט rnc/ (Range Name Create). אח"כ נשנה את {r 5} כדלקמן:

במקום המספר 5 — {r 5}

נכתוב את שם התחום NN1 — {r NN1}

בתא ששמו NN1 נכתוב את המספר 5 — NN1 [5]

שתי הפקודות בדוגמה זו מניבות תפוקה זהה - הסמן ינוע 5 ימינה. השוני ביניהן בכך שבפקודה הראשונה הארגומנט הוא מספר קבוע (5) שניתן לשינוי ידני בלבד (על ידי חדירה למקרו לפני הפעלתו) ואילו בשניה הארגומנט הוא מספר משתנה (NN1). למשל אם נכתוב 12 ב NN1 ינוע הסמן 12 פעמים ימינה. בהמשך פרק זה נלמד איך אפשר במהלך מקרו "חי" (פעיל) להחדיר נתונים באופן אוטומטי ואינטראקטיבי לתוך NN1 ודרכו אל תוך פקודת המקרו.

עד כאן ההסבר למשפט a\ בתרגיל 1.02. התקן אותו בגיליון ואח"כ שנה ערכים ב NN1 והפעל את המקרו בשני האופנים.

עכשיו במקום לכתוב מספר ב NN1 כתוב שם את הפונקציה

@cellpointer("contents")

הזז את הסמן אל תא סמוך, הקש בו מספר, הקש F9 ובצע את המקרו. אח"כ הוסף במשפט מקרו a\, לפני הפקודה {home}, את הפקודה {calc} (אוטומציה ל F9) - {calc}{home}... - וחזור על הניסוי.

נבחן עכשיו את המטלה שמבצע משפט b\ בתרגיל 1.02. נאמר שבמקום כל שהוא הקמנו תחום לקליטת נתונים בשם Data1. ב Data1 יש 3 עמודות ו 12 שורות ובכל שורה נתונים שנכתבו בעמודות מימין לשמאל. נאמר שהסמן נמצא בתא הימני העליון של Data1 ורוצים להציבו שורה אחת מתחת לשורה האחרונה של המטריצה כדי להוסיף עוד נתונים.

נאמר שהסמן בשורה הראשונה. אם נוליכו 11 פעמים למטה יתייצב בשורה האחרונה של Data1 אך אם נוליכו 12 פעמים (כמספר השורות במטריצה) יתייצב בשורה שרצינו.

בהמשך נלמד להגדיר שם תחום באופן אוטומטי ע"י מקרו ומשנעשה זאת לא נדע מראש כמה שורות יש ב Data1 ברגע נתון - אז מה עושים? נסה לפתור את הבעיה בעצמך לפני שתמשיך.

בלוטוס יש פונקציה שמחזירה (נותנת, מניבה, מפיקה) את מספר השורות בתחום שהוגדר מראש

בדוגמה הנוכחית @rows(Data1) = 12

בפעולה מספר 2 בתרגיל 1.01 כתוב {d 7}. אפשר להשתמש במה שלמדנו זה עתה ולכתוב משוואה זו ב NN1. אבל משוואה "חיה" מגדילה צריכת זיכרון ומאיטה את מהירות הביצועים. נוהל יעיל יותר שחוסך זיכרון ומחשב את הפונקציה במהירות רבה יותר מפעיל פונקציה זו מתוך משפט המקרו עצמו. לעומת זאת, משוואה חיה מגיבה לשינויים בגיליון בזמן אמת בעוד שהמשוואה במקרו מגיבה רק בעת הפעלתו. לפיכך יש בהחלט מקום להשתמש בשתי השיטות ועוד נדון בכך. ועכשיו

במקום לכתוב את המספר 7 {d 7}

נכתוב את המשוואה במקרו {d @rows(Data1)}

לכן, יהיה מספר השורות ב Data1 אשר יהיה, תמיד ינוע הסמן ממקומו הנוכחי כלפי מטה מספר שורות כמספר השורות הנוכחי ב Data1 ויתייצב בשורה הריקה הראשונה מתחת לתחום הזה.

הקם בגיליון תחום בשם Data1 ואת המשפט \b והפעל אותו בשני האופנים - בשלבים וללא שלבים.

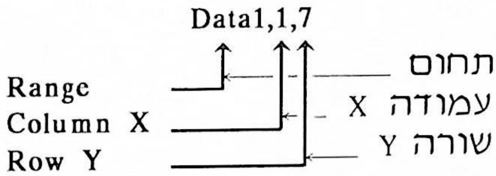
הארגומנט יכול להיות כל פונקציה שרוצים. אביא כאן שתי דוגמאות שיתרמו גם להבנת הקשר בין מבנה מטריצה לבין ארגומנטים בפונקציות.

כפי שראינו יש בתרגיל 1.02 מטריצה בשם Data1 עם 3 עמודות ו 12 שורות. מספר העמודה הראשונה משמאל הוא 0 ולכן מספר העמודה השלישית הוא 2 (ראה איור בתרגיל). מספר השורה הראשונה מלמעלה הוא 0 ולפיכך השורה השנים עשר מספרה 11. יש כאן מפה בתוך מפה (כמו משבצת במפה טופוגרפית) ולה רשת קואורדינטות פנימית משלה המאפשרת זיהוי כתובות בתוך המטריצה עצמה ללא תלות בנוהל כתובות בגיליון. כתובתו בגיליון של כל תא יכולים לכתוב בשתי צורות - ע"י אות העמודה ומספר השורה בגיליון או על ידי שם התחום, מספר העמודה בתחום ומספר השורה בו. למשל את כתובת התא המוצל ב Data1 בתרגיל זה אפשר לכתוב:

כתובת התא - בגיליון f16

או

כתובת התא - במטריצה Data1



מבנה זה של ארגומנט חשוב ביותר וממולץ ללמוד אותו היטב כי הוא מרכיב חיוני באוטומציה וישולב במקומות שונים במשפטי מקרו בהמשך. לאור זאת נבחן את הפונקציה הבאה.

```
@index(Data1,x,y)
```

פונקציה זו מחזירה את תוכן התא הממוקם במפגש בין עמודה x לבין שורה y באיזור ששמו Data1. הפונקציה נקראת אינדקס כי המספרים x ו y הם אינדקס התא המבוקש בתחום Data1.

בתרגיל 1.02 כתובה פקודת המקרו הבאה

```
{d   @index(Data1,1,3)}
```

משתופעל - כמה פעמים ינוע הסמן כלפי מטה?

תשובה: 127. מדוע?

וכתובה גם הפקודה

```
{u   @n(Data1)}
```

כמה פעמים ינוע הסמן ולאיזה כיוון?

תשובה: 87. מדוע?

הפונקציה @n(Data1) מחזירה את תוכנו של התא השמאלי העליון במטריצה כשהוא VALUE ו @s(Data1) מחזירה אותו אם הוא LABEL.

הקם את כל משפטי המקרו שבתרגיל הנוכחי בגיליון עבודה והפעל אותם פעם באופן רגיל ופעם ב Step.

הסבר את משפט המקרו \c. איזה שימושים יש לו?

> **משוואה "חיה"**
>
> פונקציה בתא בגיליון תופסת מקום בזיכרון וזמן חישוב רב אבל מגיבה בזמן אמת

החלף את מספרי השורה והעמודה בפונקצית האינדקס שבמשפט \e ,
רשום מספר כל שהוא בתא שהוגדר ע"י הקואורדינטות החדשות ב Data1
והפעל שוב את המשפט \e .

האם אפשר לכתוב כארגומנטים x ו y ב @index משתנים דינמיים?
לשם מה? נסה בעצמך.

הוסף שני תחומים לגיליון. לאחד קרא Col1 (ע"י /rnc) ולשני Row1. מלא את כל התאים של Data1 במספרים ו\או בטקסט וכתוב בתא ליד המטריצה את הפונקציה

@index(Data1,Col1,Row1)

הכנס מספרים שונים ב Col1 וב Row1 וראה כיצד משתנה תפוקת המשוואה (החיה) כל פעם.

מה קורא כשכותבים ב Col1: ?@cols(Data1)-@cols(Data1)? מדוע?

מה קורא כשכותבים ב Col1: ?@cols(Data1)-1? מדוע?

מה קורא כשכותבים ב Col1: ?@cols(Data1)+1? מדוע?

מה קורא כשכותבים ב Row1: ?@rows(Data1)-@rows(Data1)? מדוע?

ערוך את הניסוי הבא.

ב Data1 בעמודה הראשונה מימין הקש שמות משפחה, בעמודה האמצעית שמות פרטיים ובראשונה משמאל כתובות - בכל שורה.

הכן בגיליון טופס קטן לקליטת נתונים מ Data1 בן שתי עמודות ושלוש שורות. בעמודה הימנית יהיה תאור הפריט ובשמאלית פונקציה לפי הציור הבא. לפני הקלדת הפונקציות בצע /rncRow1 .

תזכורת להפעלת מקרו בשלבים - Step

1. הקשה על צמד המקשים Alt F2
2. הפעלת המקרו: הקשה על המקש Alt יחד עם אות המקרו
3. הקשה איטית חוזרת על מקש כל שהוא עד לסיום
4. לביטול מצב זה מקישים שנית על הצמד Alt F2

		תאור הפריט	שליפה מבסיס הנתונים
		==================	==================
☐	Row1	מ ש פ ח ה	@index(Data1,2,Row1)
		פרטי	@index(Data1,1,Row1)
		כ ת ו ב ת	@index(Data1,0,Row1)

עכשיו הקלד ב Row1, לאט, מספרים מ 0 ועד 11 וראה את המתרחש בגיליון. לפניך שליפת נתונים ע"י הולכת סמן לוגי. בהמשך נלמד כיצד לבצע זאת בעזרת משפט מקרו פשוט, מהיר וידידותי באופן אוטומטי לגמרי תוך ניצול מירבי של מהירות המחשב והגיליון.

מה ההבדל בין שלושת המשוואות בטופס שלמעלה? נסה לחשוב על "פטנט" שיאפשר שימוש במשוואה אחת במקום שלוש. את העיקרון ניתן לגלות כבר עכשיו, הפתרון השלם משלב משוואה במקרו ונלמד אותו עוד מעט.

ולמה הדבר דומה. הרי זה כאילו כתבנו בתא אחד את הפונקציה

@cellpointer("contents")

ואחר כך "טיילנו" עם הסמן על פני המטריצה כשאנו מקישים {calc} - F9 - כל פעם לעידכון הגיליון. כלומר, למעשה ביצענו "כאילו" הולכת סמן. ובמילים אחרות

הולכת סמן לוגי - Virtual Cursor Movement

בנושא זה נטפל בהרחבה בדפים הבאים.

1.2.1.2 סמן בפועל - תנועה לא סידרתית

איור 1.04, בחלקו העליון, מתאר בצורה גרפית את מסלול תנועת הסמן כפי שלמדנו עד כה. תנועה זו מתבצעת על ידי כך שהסמן עובר תמיד מכתובתו הנוכחית לתא שכן הצמוד אליו מעליו או מתחתיו, מימינו או משמאלו. דהיינו גרף תנועת הסמן מצוייר רק בקוים ניצבים כשהוא מונע במקשי החיצים או חלופתם בפקודות המקרו.

מאפין חשוב של תנועה זו הוא שכדי להגיע אל תא מרוחק מכתובתו הנוכחית חייב הסמן לעבור כל אחד ואחד מהתאים ביניהם ומסלולו מתווה גראף של קוים ניצבים. לתנועה זאת אקרא

תנועה סידרתית ניצבת - Perpendicular Sequential Movenent

אל תהסס לערוך כל ניסוי שעולה בדעתך

וזאת משום שהיא מתרחשת כך - הגעתו של הסמן אל כתובת היעד מותנית בכך שבשלב אחד בלבד קודם לכן היה בתא סמוך. נבחן עתה מה זה סמן ומדוע מקשי החיצים מניעים אותו בצורה מגושמת ואיטית כזאת.

הסמן הוא מלבן בוהק בתאו ואפשר להניעו מתא אל תא. כלומר הוא קיים פיזית, בפועל, וניתן לראותו בעין. אלא שאין תנועה זו דומה כלל להולכת קרן אור מפנס על פני המסך. יותר דומה היא לכיבוי אורו של הסמן בתאו והדלקתו מיד בתא סמוך. אפשר אולי לאמר שמכבים את הקואורדינטות של תאו הנוכחי ומדליקים מיד את אלה שבתא הסמוך.

אם כך אפשר לכבותו בתאו ולהדליקו בתא אחר המרוחק ממנו מרחק רב ואז ינוע בין תאים מרוחקים ככל שיהיו במהירות זהה למהירות תנועתו בין כתובות סמוכות. תנועה לא סידרתית זו מתבצעת ע"י המקש - F5 - GoTo.

פקודת המקרו לכיבוי הסמן בכתובת המוצא והדלקתו מיד ביעד היא

{goto}c15~

התו Tilde ------------------> ~

בטקסט של משפט מקרו מורה למחשב להפעיל את המקש ENTER.

בפקודה זו מלת ההפעלה מוקפת בסוגריים מסולסלים בעוד הארגומנט נכתב מימין. אם נכתוב

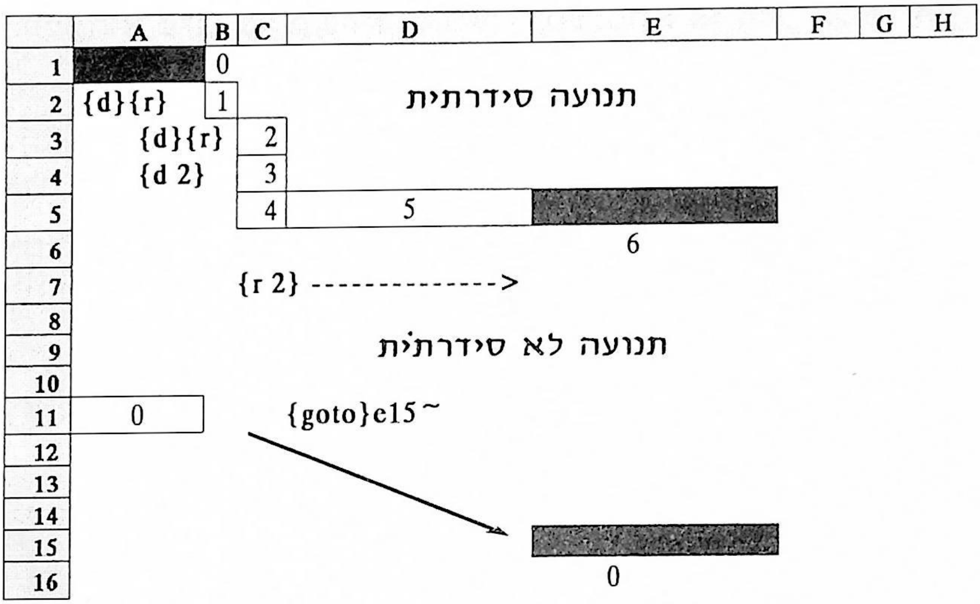
{goto}Data1~

ינוע הסמן מכתובתו הנוכחית אל מקומו של התא העליון משמאל בתחום Data1. בהמשך נלמד שיטה מיוחדת לשילוב משתנה דינמי בארגומנט של פקודה זו.

- מקרו כותבים בעמודה אחת בלבד
- אפשר להמשיכו תא מתחת לתא
- בסוף פיסקת המקרו כתוב {quit}

- אם שם תחום זהה לכתובת תא - לכתובת התא יש קדימות.
- אפשר לצרף אותיות למספרים בשם תחום - צריך לזכור לבחור צימדי אותיות מסוף הא"ב - נוהל זה נוח לאירגון חיבור המקרו ולשמירה על אחידות שמות בקבצים שונים.

איור 1.04
{goto}

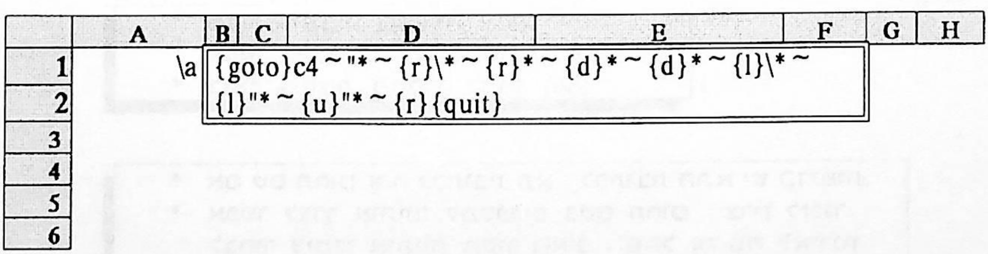


בתרגיל 1.2 הוסף והפעל את משפט המקרו

\f {home}{goto}Data1~{r 2}{d @rows(Data1)}{quit}

מה ההבדל בין המשפט הזה לבין המשפט \b באותו תרגיל?

תרגיל 1.03
ה ק ל ד ת   ת ו י ם   ב מ ק ר ו



תרגיל 1.04
הקלדת משוואות במקרו

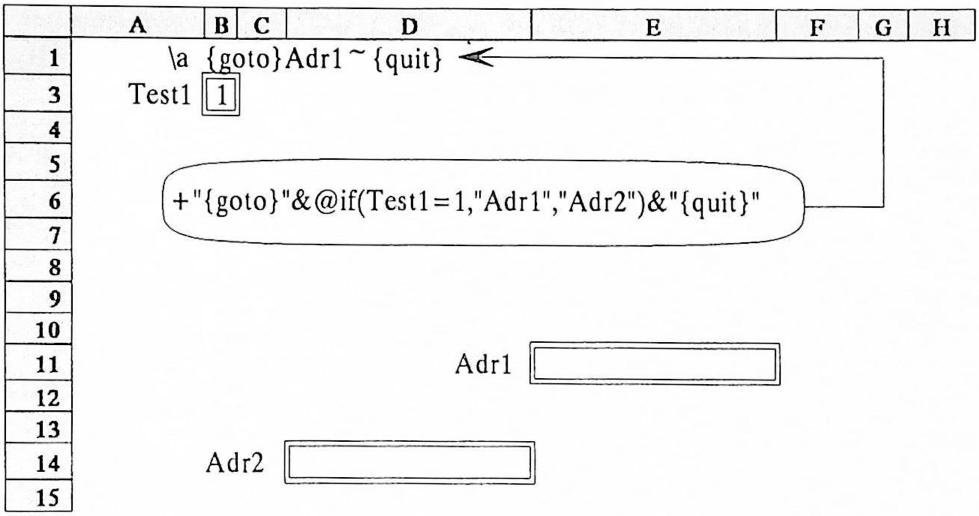
	A	B	C	D	E	F	G	H
1	\b {goto}d4~@sum({d 2}.{end}{d})~/c~{r}~{quit}							
2								
3								
4						סה"כ		
5				====	====			
6				25	165			
7				37	215			
8				38	535			
9				39	202			
10				121	399			
11				576	782			
12				1	174			
13				12	212			

העתק את שני התרגילים, 1.03 ו 1.04 והפעילם בשלבים (Step) ובדרך הרגילה. בעת ההפעלה בחן בתשומת לב כיצד מתבצעת הקלדת הנתונים במקרו. שינויים מתחוללים לעיניך בשלושה מקומות בו-זמנית: מעל הסרגל העליון, בגיליון ובפינה השמאלית מתחת לגיליון. בצע לאט ומספר פעמים.

יש כאן מספר מרכיבים חדשים. נסה לפענח אותם בעצמך בעזרת הטכניקה של רישום מספר סידורי רץ לכל פעולה - סדר פעולות. בטכניקה זו נשתמש אח"כ כשנלמד דרכים להגדרת צרכים לפני שניגשים לכתוב את חיבור המקרו.

בספר זה ננסה לפתח חשיבה יוצרת ולימוד עצמי. לשם כך יכללו במשפטי המקרו מפעם לפעם אלמנטים חדשים שיוסברו בהמשך. אבל אם גילית את העיקרון אל נא תהסס "לשחק" ולנסות. למשל בתרגיל 1.04 שולבה הפעלת פקודת Copy במשפט המקרו.

תרגיל 1.05
משוואה ל {goto}



ראינו מקודם שהארגומנט של {goto} נמצא מחוץ לסוגריים המסולסלים מצד ימין ומיד אחריו מופיע התו ~ , ממש כאילו ביצענו זאת ידנית. נשאלת השאלה כיצד לשלב ארגומנט דינמי בפקודה זו. שיטה אחת נלמד בתרגיל 1.05.

נאמר שיש בגיליון שלשה תאים שקיבלו את השמות Test1 Adr1 Adr2. רוצים לכתוב משפט שיציב את הסמן באחת משתי הכתובות Adr1 או Adr2. באיזו מהן יוצב הסמן יקבע מספר שיכתב בכתובת Test1. אם כתוב שם המספר 1 רוצים שהסמן יוצב ב Adr1. אחרת - אם הוא איננו 1 - רוצים שיוצב ב Adr2. לשם כך נכתוב את משפט המקרו לא כ LABEL אלא כמשוואה. בתרגיל זה רואים בשורה 6 את המשוואה שלמעשה כתובה בתא b1 וב b1 רואים את תפוקת המשוואה. דהיינו אם נכתוב את המשוואה הזאת בתא b1 נראה בעין את מה שרואים בו עכשיו. נבחן עתה את סדר הפעולות של משפט המקרו הזה.

הדברים שלמדנו עד כה יהיו אחר כך מרכיבים חשובים ביישומי מקרו שונים. למשל, מקרו לניהול אוטומטי של בסיס נתונים מקרו לשליטה אוטומטית בגיליון ומטלות רבות אחרות.

כמו כן יש צורך להבין היטב את כל מה שנלמד עד כה לפני שעוברים ללמוד הפעלת מקרו בלולאות - בהמשכו של ספר זה.

נבחן עתה צעד אחר צעד את סדר הפעולות של המשפט הבא.

משפט המקרו

+"{goto}"&@if(Test1=1,"Adr1","Adr2")&"{quit}"

סדר הפעולות

1. + — לידע את המחשב שמדובר במשוואה
2. +"{goto}" — משוואה זו מחזירה {goto} בתא בו היא כתובה
3. & — הסימן הזה מחבר מחרוזות טקסט (concatenation) והוא נקרא אמפרסנד (ampersand). למשל, נכתוב ב C את המלה - רותי. ב B נכתוב - רמי אוהב את וב A את המשוואה - "&B" +C&", נקבל - רמי אוהב את רותי (" " נותן תו רווח בין שני חלקי המשפט שחיברנו).
4. @if(Test1=1,"Adr1","Adr2") — אם Test1 = 1 אז Adr1 אחרת Adr2
5. &"{quit}" — סוף המשפט

תרגיל זה רצוי לבצע לפי הסדר הבא. הגדר בגיליון את כל שמות התחומים הדרושים לתרגיל. לאחר מכן כתוב בתא כל שהוא את המשוואה שבפעולה מספר 4 . הקש מספרים שונים ב Test1 וראה כיצד משתנה תפוקת המשוואה. אח"כ כתוב את משוואת המקרו ב b1 ושוב הכנס מספרים שונים ב Test1. הפעם ראה כיצד משתנה משוואת המקרו. לבסוף הפעל את המקרו בשלבים וללא שלבים, כל פעם עם מספר אחר ב Test1. עתה בצע את נוהל "רמי אוהב את רותי" כמוסבר בפעולה מספר 3.

1.2.1.3 סמן בפועל - תרגיל מסכם

אחד הדברים שעשינו עד כה מבלי להתייחס אליו במפורש הוא אירגון הגיליון לפני כתיבת מקרו. משפט מקרו פועל על דברים שנמצאים בכתובות שונות ובאיזורים שונים בגיליון. אם לא נלמד כיצד לארגן את כל הגיליון לא נפיק את מירב התועלת מהלימוד הזה. עד כה נוכחנו שיש שני מרכיבים חשובים בחיבור מקרו: פעילות שוטפת ונראית לעין (כמו קליטת נתונים) וקבוצה של משפטי מקרו. את איזור הפעילות הגלויה נתכנן ונקים בעמודות הראשונות משמאל ליד הסרגל השמאלי - החל בשורה הראשונה. נשתמש בכמה עמודות שצריך ואח"כ נשאיר עמודה פנויה. שתי העמודות הבאות יוקדשו לאיזור המקרו. בעמודה הימנית שבאיזור זה נכתוב את משפטי המקרו ונגדיר את התחומים. בעמודה שמשמאל נרשום לתזכורת את שמות המקרו ואת שמות התחומים. ראה והשווה את אותיות העמודות באיור 1.05 ובאיור 1.06. פרט לחריגים נדירים לא יהיו שני איזורים אלה באותן עמודות.

איור 1.05
הכנת מבנה הנתונים לפני כתיבת מקרו

	A	B	C	D	E	F	G	H
1				לוח 1.1				
2								
3				תקציב בשקלים לפי סעיף ויחידה				
5								
5								
6								
7	XX11			סעיף תקציבי				יחידה
8	======	==	==	============	============	======	==	===
9	5	4	3	2	1	סה"כ	שם	#
10	======	==	==	============	============	======	==	===
11	80,442	0	0	338,534	537,326		סה"כ	
12	----------	---	---	----------------------	-------------------	-------		------
13			Total1					
14	3,471	0	0	5,364	22,553	31,388	א	1
15	25,671	0	0	32,495	78,483	136,649	ב	2
16	13,800	0	0	235,151	206,156	455,107	ג	3
17	37,500	0	0	65,524	230,134	333,158	ד	4
18								
19	Data1							

איור 1.06
מקרו לאיור 1.05

	J	K	L	M	N	O	P	Q
1	\a {1}' <-- /rnc ע"י XX11 לתא זה קראנו בשם~{quit}							
2	\b {1}Data1 הסמן בפינת~{goto}Data1~{quit}							
3	\c {1}Total1 הסמן בפינה של~{goto}Total1~{r 5}{quit}							
4	\d {1}'שורה חדשה~{goto}Data1~{end}{r}{end}{d 2}{quit}							
5	\e {1}'מספר רץ~{goto}Data1~{end}{r}{end}{d 2}1+{u}~{l}{quit}							
6	\f {1}'משוואה~{goto}Total1~{r 5}@sum({l}.{end}{l})~{quit}							
7	1 {home}{blank XX11}~{goto}XX11~{return}							

> במשפט מקרו אפשר לשלב כל תו בעברית ו/או באנגלית.

בתרגיל זה מתנהל דו-שיח יזום בינינו לבין המחשב. התרגיל בנוי

מסידרת משפטים שכאשר נפעילם בסדר הא"ב "נטייל" במקומות שונים באיזור הפעילות הגלויה. כמו בכל טיול רוצים לקבל הסבר בכל מקום שמגיעים אליו. בעת טיול המקרו יופיע הסבר קצר בתחום ששמו XX11 כל פעם שנשנה מקום.

בתחילת כל מקרו נכתוב משפט (זהה בכל שש המשפטים) שימחק את ההודעה הקודמת מ XX11 ויוליך לשם את הסמן לפני כתיבת ההודעה החדשה. אולם כדי לא לחזור על הקטע הזה 6 פעמים (שהרי הוא זהה בכולם) נפעיל את עקרון ה ...

מ ו ד ו ל ר י ו ת

נבצע זאת על ידי קריאה למשפט שירות.

את משפט המקרו המשותף לכולם נכתוב רק פעם אחת, בתחילת כל אחד מהמשפטים הראשיים, ונקרא לו בשם 1 (כן, אפשר לתת מספר כשם של מקרו). ומכיוון שהוא "משרת" מקרו אחרים נקרא לסוג זה של משפט: "משפט שירות". נבחן עכשיו את סדר הפעולות שלו.

מ ש פ ט   ה ש י ר ו ת

1 {home}{blank XX11}~{goto}XX11~{return}

ס ד ר   ה פ ע ו ל ו ת

1 {home}

2 {blank XX11}~ — ימחק תוכן התחום XX11.
(כמו re/ רק יותר יעיל)

3 {goto}XX11~

4 {return} — חלק מנוהל קריאה למשפט שירות.

תוים מפרידים בין ארגומנטים
במשוואות
ארגומנטים בסוגריים מופרדים ע"י פסיקים. מלת הפעלה לפני הסוגריים.
בפקודות מקרו
רווח בין מלת ההפעלה וארגומנט ראשון ופסיקים בין הארגומנטים

משפט שירות הוא בדיוק כמו משפט מוסגר בשפת בני אדם. במהלך המשפט הראשי פותחים סוגריים (לאמור, כאן מתחיל משפט חריג)

ובסיומו סגירת הסוגריים - הוראה להמשיך ולקרוא את התוים בדיוק מאותו תו שהיינו מגיעים אליו אם לא היה המשפט המוסגר קיים.

סוגריים פותחים (קריאה למשפט שירות) ע"י כתיבת שם משפט השירות בתוך סוגריים מסולסלים. בדוגמאות שלמעלה: {1}. למשל,

המשפט הראשי

\b {1}Total1 הסמן בפינת~{goto}Total1~{r 5}{quit}

1 {home}{blank XX11}~{goto}XX11~{return}

משפט השירות

סדר הפעולות

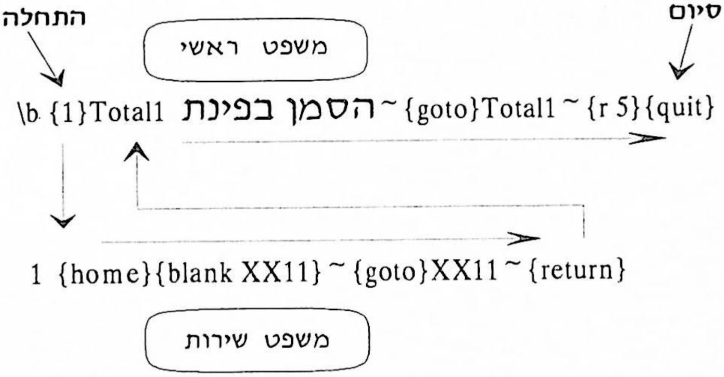
1 {1}	קריאה למשפט שירות ששמו 1
2 {home}	פקודה ראשונה במשפט השירות ששמו 1
3 {blank XX11}~	
4 {goto}XX11~	
5 {return}	פקודה אחרונה במשפט השירות ששמו 1
6 Total1 הסמן בפינה של~	אחרי חזרה ממשפט השירות
7 {goto}Total1~	
8 {r 5}	
9 {quit}	פקודה אחרונה ב \b

הפקודה {return} בסוף משפט השירות מורה למחשב לחזור להמשך המשפט הראשי - כאן יקרא המחשב את ה T שבמלה Total{1}.

ובצורה גראפית אפשר לתאר את המעברים הללו כך:

> **משפט שירות תמיד מסתיים**
> **ב {return} ואך ורק ב {return}**

סדר פעולות במערך מודולרי



בשפת דיבור משתמשים בסוגריים במשורה ועל אחת כמה וכמה בסוגריים בתוך סוגריים. וזאת משום שקשה לזיכרון האדם לעקוב אחרי הקשר הדברים כשיש סטיות רבות מהנתיב המרכזי או שרשרת ארוכה של סוגריים בתוך סוגריים (לשרשרת כזאת במחשב קוראים Nesting). למחשב אין מגבלה כזאת ולכן יש כאן שוני של ממש בין שפת אנוש לשפת מחשב.

ראינו עד כה כיצד הולך ונרקם חיבור מקרו וראינו חלק ממרכיביו כגון - פקודת מקרו, משפט מקרו, מבנה מוכן בגיליון והקשרים ביניהם. למדנו נוהלים, כללי כתיבה וטכניקות שונות. עם הקריאה למשפט שירות (השימוש בסוגריים) נכנסים חזק לעומקה של שפת המקרו. עד כה למדנו מקרה פשוט, אך בסיסי, אחד ובמהלך כל הפרקים הבאים נפתח יותר ויותר את השימוש בעיקרון המודולריות ונראה מגוון רחב ביותר של טכניקות פשוטות ליישומו.

עיקרון המודוליות כפי שהוא מתבטא בנוהל משפט שירות פותר בעיה של קטעי משפטים שחוזרים בתוך משפט אחד ו\או משפטים שונים. נוהל משפט שירות ממלא תפקיד נוסף, וחשוב ביותר, שנלמד אותו בתרגיל הבא.

תרגיל 1.06
משפט שירות ל {goto}

	A	B	C	D	E	F	G	H
1	\b	{goto}{AdrX}~{quit}						
3	AdrX	Adr1{return}		< ----- @if(Test1=1,"Adr1","Adr2")&"{return}"				
4	Test1	1						
5								
6								
7				Adr1				
8								
9		Adr2						

בגיליון חדש הקם את ארבעת התחומים Test1 Adr1 Adr2 AdrX. בכתובת AdrX כתוב את המשוואה מ AdrX שבתרגיל 1.06. שנה את הערכים ב Test1 וראה כיצד משתנה תפוקת המשוואה שזה עתה כתבת. תפוקה זו היא משפט שירות (מדוע?) בנוי כמשוואה. לפיכך מתעדכן משפט שירות זה בזמן אמת כל פעם שמשתנה הגיליון. עכשיו אפשר לקרוא ל AdrX מכל מקום בפיסקת המקרו כמה פעמים שרוצים ואין צורך שפיסקת המקרו עצמה תכתב כמשוואה גדולה ומסובכת. הבה נראה את סדר הפעולות.

1 {goto} — פקודה ראשונה במשפט הראשי ששמו \b
2 {AdrX} — קריאה למשפט שירות ששמו AdrX

3 Adr1 — מלה ראשונה במשפט השירות - אם Test1 = 1
4 {return} — חזור למשפט הראשי (סגור סוגריים)

5 ~ — ENTER - פקודה ראשונה ב \b אחרי סגירת הסוגריים
6 {quit} — סיום משפט המקרו

כל אימת שנכתוב {AdrX} במשפט המקרו משתלבת באותו מקום תפוקתו של משפט שירות זה. אבל משום שמשפט זה הוא משוואה חיה הוא מגשר בין השינויים שהמקרו מחולל בגיליון לבין הפקודות בהמשכו. מקרו לומד זה מתאים את תגובותיו העתידיות לתפוקותיו הנוכחיות.

נאמר שהקמנו בסיס נתונים (ראה פרק ב). את איזור הנתונים הגדרנו בשם התחום Data1 (מסגרת כפולה בתרגיל). ואח"כ הוספנו שורה חדשה עם שם חדש מיד אחרי הגבול התחתון של Data1 (קו שבור). הוספת השורה החדשה התבצעה אוטומטית במהלך פיסקת מקרו.

תרגיל 1.07
הגדרת תחום אוטומטית



כתוב פיסקת מקרו להגדרה אוטומטית של התחום Data1 כאשר ידוע לפני הפעלת הפיסקה כי: 1. קיים תחום בשם Data1; 2. התחום הזה מכיל לפחות שורה אחת; 3. מספר השורות ב Data1 לא ידוע מראש; 4. מספר העמודות ב Data1 ידוע מראש.

לכאורה פשוט למדי. כדי להגדיר תחום במקרו כותבים במשפט את רצף התוים ~/rncData. בעת הפעלת המקרו יבין המחשב שנדרש ממנו לבצע את הפעולות הבאות:

קרא לתפריט הפקודות של הגיליון		1. /
בחר משם את הפקודה	Range	2. r
אח"כ בחר באופציה	Name	3. n
והפקודה הבאה תהיה	Create	4. c
כתוב את שם התחום להקמה		5. Data1
ENTER		6. ~

ומקבלים /rncData1~

כדי לבצע את הקמת התחום נתנו את הפקודה /rnc , סיפקנו את שם התחום Data1 ועתה אנו מתבקשים לספק את כתובת התחום בשטח.

אילו ידענו שבתחום הנוכחי יש יותר משורה אחת יכולנו לכתוב

```
\a  /rncData1~{d}~{quit}
```

ואילו ידענו שבתחום מוגדר תא אחד בלבד היינו כותבים

```
\b  /rncData1~.{d}~{quit}
```

שים לב לנקודה שנוספה במשפט המקרו

הקם בגיליון תחום בשם Data1 על יותר מעמודה אחת ושורה אחת ואת שני משפטי המקרו שלמעלה. (אם לא הצלחת לכתוב את המשפט ובמקום זה קיבלת את תפריט הפקודות זה סימן ששכחת להתחיל את הרצף עם גרש - פסיק עליון.)

הפעל כל אחד מהמשפטים הללו פעם אחת בשלבים ופעם ללא שלבים. אח"כ הגדר מחדש את Data1 בתא אחד בלבד והפעל שנית את שני המשפטים בשני האופנים.

איזה משני המשפטים עונה על כל האפשרויות ומדוע?

הבה נוסיף לתרגיל מימד נוסף ומענין. חלק ניכר מבסיסי נתונים בגיליון האלקטרוני אפשר לבנות ממספר קבוע של עמודות אבל ממספר שורות משתנות.

ובכן, נתון בסיס נתונים שיש בו מספר ידוע מראש של עמודות ואפשר שיהיו בו עמודה אחת או יותר. אבל לא יודעים כמה שורות נוספו - אחת, יותר מאחת או אף אחת. ידוע שיש (מהגדרת התחום בשלב קודם) שורה אחת לפחות.

הקם בגיליון מספר תחומים שידגימו את האפשרויות הללו והכנס בהם נתוני דמה. נסה לפתח את משפט המקרו \b ולהוסיף לו רצף תוים שיענו גם על הנתונים החדשים הללו.

אחת הטכניקות להולכת סמן מהירה היא גלישה לאורך רצף תאים עם נתונים (או ריקים). אחרי הצבת הסמן בתחילת הרצף מקישים End ואח"כ מקש החץ המורה את הכיוון המבוקש. אפשר לשלב טכניקה זו באוטומציה של הגדרת התחום אם נכתוב, למשל

```
\b  /rncData1~.{end}{d}~{quit}
```

שים לב לנקודה במשפט זה

כאן מתעוררת בעיה. שהרי איננו יודעים מראש אם נוספו שורות ואם נוספו איננו יודעים כמה נוספו.

בצע את התרגיל הבא. הוסף שורה לאחד מתחומי הנתונים שהקמת במהלך תרגיל זה. שנה את \b לפי הדוגמה האחרונה והפעל אותו ב ש ל ב י ם ב ל ב ד על התחום הזה. עכשיו, שוב בשלבים, הפעל אותו שנית מבלי שבצעת שינויים כל שהם בנתונים. דהיינו - לא נוספה שורה חדשה. מה ההבדל בין שני המקרים? מה הפיתרון?

ההבדל הוא בכך שכאשר נוספה לפחות שורה אחת חדשה יש לסמן רצף לגלוש עליו ולעצור בסופו ולכן תוגדר נכון כתובת התחום. אך משלא נוספה אפילו שורה חדשה אחת הרצף לגלישה הוא הריק שמשתרע בין הנתונים ותחתית הגיליון ולכן יגלוש הסמן עד למטה וכך גם תתקבל הגדרה שגויה לתחום.

עתה הבה ואציב בפניך אתגר. הפיתרון כתוב כאן בעוד כמה שורות. אנא, סגור את הספר ונסה למצוא פיתרון משלך לפני שתמשיך לקרוא.

הפיתרון נמצא במשפט המקרו הבא וללא הסברים נוספים, נסה ותיווכח.

```
\b          /rncData1~.{end}{d   2}{end}{u}~{quit}
```

מדוע {d 2}?

מסתבר שהרצף {end}{d 2}{end}{u} נדרש במקומות רבים במשפטי מקרו רבים. אפשר לומר שזה רצף "אוניברסלי" והמודולריות מנחה אותנו מיד להפוך רצף זה למשפט שירות אוניברסלי. מאחר שיש ביקוש למשפט הזה בקבצים רבים נכלול אותו בעתיד גם בסיפרית המקרו שלנו.

אני נוהג לכתוב פיסקה זו כך

```
\b   {goto}Data1~/rncData1~.{31}~{quit}
31   {end}{d 2}{end}{u}{return}
```

1.2.2.1 סמן לוגי - פקודת /c

נייחד שוב כמה מילים לסמן. אפשר להגדיר את הסמן, כמו כל דבר אחר בעולם, בשני מובנים. במובן הפיזי זה מלבן בוהק בתוך תא שאפשר להוליכו לתא אחר בהקשה על קלידים במקלדת. אפשר גם לבחון את הסמן מנקודת הראות של הפעולות שהוא מבצע, האפקטים שהוא יוצר.

האפקט שיוצר הסמן הוא סימון חד ערכי של תא כל שהוא - או קבוצת תאים בצביעת שטח. מטבע הדברים מסמנים כתובות לפני שניתן לבצע פעולות כל שהן כגון העתקה, כתיבה בתא, העברה, מחיקה, הוספת או ביטול שורה וכיוצא באלה. ברגע שיצרנו נוהל סימון כתובות עם אפקט זהה לסמן בפועל אך ללא הקשה על קלידי החיצים וללא הפעלת חלופתם במקרו יצרנו סמן לוגי.

vir-tu-al existing in essence or effect
though not in actual fact or form

The American Heritage Dictionary

דוגמה לכך היא הפקודה Copy. סדר הפעולות המוליך לתוצאה המבוקשת:

1. /		קרא לתפריט הפקודות של הגיליון
2. c	Copy	בחר משם את הפקודה
3. prompt = where from?:		הקש כתובת המוצא
4. ~		ENTER
5. prompt = where to?:		הקש כתובת היעד
6. ~		ENTER

התפוקה זהה למה שהיה מתקבל משילוב Copy עם הפעלת קלידים. רק שבפועל לא הופעלו קלידי הולכת סמן ולא נראה פיזית סמן מתרוצץ על פני הגיליון. במקרו יראה רצף התוים כך

/cAdr1~Adr2~{quit}

או

/cm15~Adr2~{quit}

או

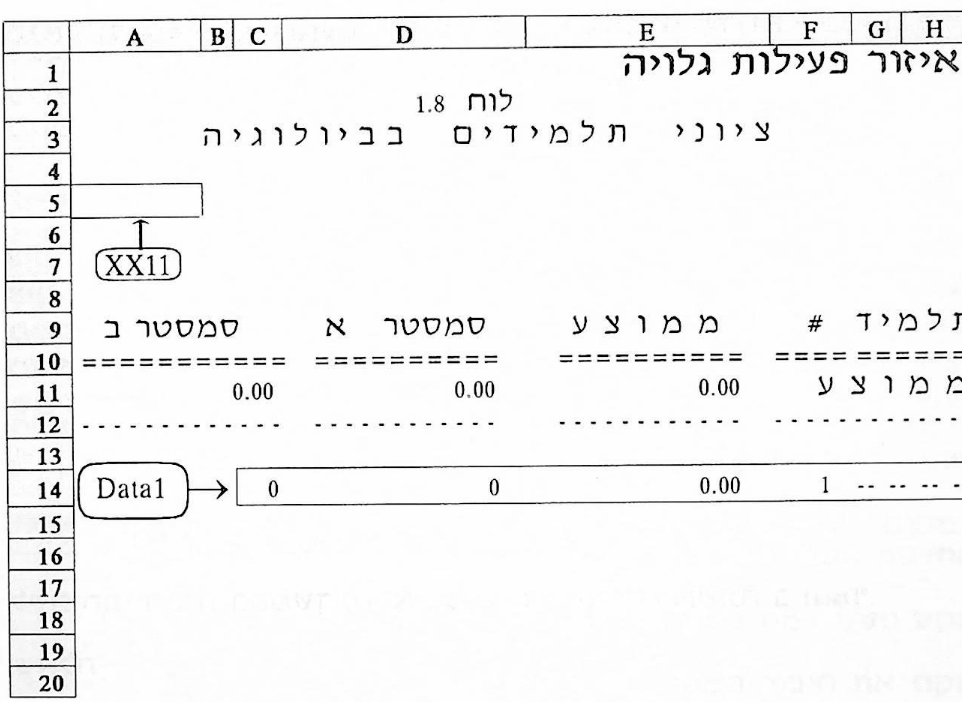
/cAdr1~~{quit}

או כל צירוף חוקי אחר בפקודה זו.

וסדר הפעולות הוא

קרא לתפריט הפקודות של הגיליון		1. /
בחר משם את הפקודה	Copy	2. c
כתובת המוצא		3. m15
ENTER		4. ~
כתובת היעד		5. Adr2
ENTER		6. ~
נקודה		7. {quit}

תרגיל 1.08
מקרו ל Copy בבסיס נתונים



למדנו כבר שכל פעילות בגיליון מחייבת מידה מסוימת של תכנון מראש. ועל אחת כמה וכמה כשרוצים להפעיל את הגיליון בנוהלים אוטומטיים. נראה עתה כיצד הוקם המבנה בתרגיל 1.08 .

מרצה לביולוגיה מלמד שני סמסטרים. בסוף כל סמסטר מתקיימת בחינה וכל תמיד מקבל ציון מ 0 ועד 100. הציון הממוצע לשני הסמסטרים הוא הציון השנתי שירשם בתעודה. בנוסף לכך רוצה המרצה להעריך את רמת הכיתה בכל סמסטר (ע"י הממוצע) וגם

לשנה כולה. הרישום הראשוני של התלמידים איננו מתחיל ונגמר בפעם אחת אלא עשוי להמשך על פני מספר שבועות בתחילת הסמסטר הראשון. לשם כך הקים הפרופסור לוח במבנה פתוח כפי שרואים בתרגיל. תחילה הגיעו 5 סטודנטים והמזכירות הודיעה שיצטרפו עוד כ 30 . בשלב ראשון הוקם הלוח כשהוא מוכן לקלוט נתונים. השורה הראשונה שנועדה לקלוט את נתוני התלמיד הראשון הוגדרה ע"י ~/rncData1 בשורה מספר 14 . העמודה הראשונה נועדה לקלוט את שם התלמיד אבל מכיוון שהראשון עדיין לא נרשם נכתב בה קו שבור. התא שמשאל - g14 - נועד לייצר מספר רץ באופן אוטומטי. לפיכך רשם בה המתכנן את הנוסחה g13+1. משמאל בעמודה f יחושב הממוצע על ידי נוסחה.

את הממוצע שיכתב בתא f14 רוצים לחשב עד לרמה של שתי ספרות משמאל לנקודה ושפורמט התא יראה זאת תמיד. לפיכך הנוסחה היא @round(@avg(d14.e14),2) והתא יפורמט ידנית בפקודה ~/rff2~ . בשתי העמודות הבאות ירשמו ציוני הבחינה בפועל במספרים שלמים ולכן לא יפורמטו. בצורה פורמלית יותר הרכב שורה מספר 14 ב Data1 נראה כך

כותרת	פורמט	תוכן	כתובת
ציון סמסטר ב	רגיל	0	d14
ציון סמסטר א	רגיל	0	e14
ממוצע שנתי	/rff2	@round(@avg(d14.e14,2)	f14
מספר רץ	רגיל	g13+1	g14
שם התלמיד	רגיל	-- -- --	h14

d14	e14	f14	g14	h14
0	0	@round(@avg(........	g13+1	-- -- --

לשם התייחסות בהמשך נקרא לשורה זו: "שורה ראשונה ב Data1".

שאלה

לפניך שתי פונקציות, א ו ב, איזו מהן מחזיקה יותר מקום בזיכרון?

א @avg(d14.d15)

ב @avg(d14.d8000)

תשובה: ההבדל היחידי בין שתי הפונקציות הוא גודל התחום שיכלול נתונים עליהם יחושב ממוצע. מאחר שגודל זה מוגדר על ידי רישום שתי כתובות בלבד בכל אחת מהפונקציות הללו אזי לכל ענין ודבר הן מחזיקות כמות זיכרון זהה.

לעובדה זו חשיבות מרובה בניהול בסיסי נתונים בגיליון. נניח שצריכים לקלוט מספרים בתאי עמודה מסוימת. בראשה נכתוב את פונקצית הממוצע ונגדיר אותה מהתא הראשון ועד תחתית הגיליון פחות כמאה שורות (כדי שאם תתבצע הוספת שורות לא תיפגע הגדרת תחום החישוב). עכשיו, כל פעם שנקיש נתון בתא כל שהוא בעמודה זו נקבל אוטומטית ובזמן אמת את הממוצע החדש בראש העמודה. לכן נכתוב במקום המיועד לחישוב הסכום בראש עמודה d את פונקציה ב ונעתיק אותה לשתי שכנותיה מימין.

עשה לך הפסקה עכשיו והעתק את התרגיל לגיליון חדש. רוחב טור לעמודות נקבע כאן באופן שרירותי - אצלך תוכל להתאימו לנתונים בצורה יותר הגיונית.

ניגש ללמוד את נוהל Line1.

איננו יודעים מראש כמה תלמידים יהיו ולכן אין טעם למלא סתם שורות ריקות במשוואות על ידי העתקת השורה הראשונה ב Data1 מספר פעמים כלפי מטה. נוהל נכון יהיה להעתיק את שורת המשוואות הזו פעם אחת בלבד כל פעם שנוסף תלמיד. אבל מהיכן נעתיקה, הרי בשורה הראשונה יש כבר נתונים? לשם כך נבצע עכשיו את הפעולות הבאות.

העתק את כל השורה הראשונה של Data1 אל השורה הראשונה בעמודה כל שהיא בסמוך ומימין לאיזור המקרו (ראה איור 1.07).

במקום החדש הזה הגדר את כל חמשת העמודות בשורה האחת החדשה בשם התחום Line1.

הקש נתוני דמה בשורה הראשונה של Data1.

הקם את חיבור המקרו לפי איור 1.08.

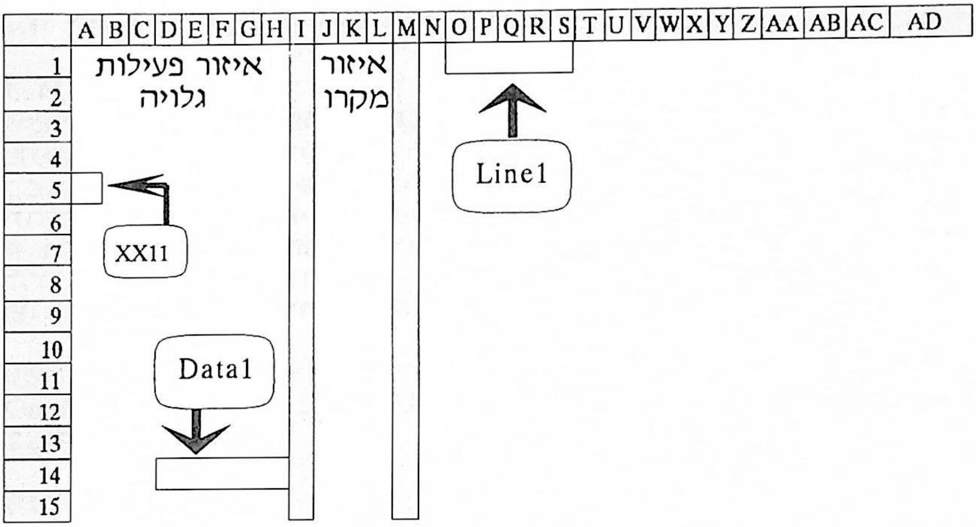
הפעל את המקרו בשלבים וללא שלבים.

הסבר את המשמעות האינטראקטיבית - אדם מחשב - שבנוהל MsgN כאשר N = כל מספר.

כתוב בפרוט מלא את סדר הפעולות בחיבור זה.

אילו יתרונות נוספים שלא הוזכרו כאן יש לנוהל Line1?

איור 1.07
אירגון הגיליון לתרגיל 1.08



איור 1.08
מקרו לתרגיל 1.08

```
\a   {home}{goto}Data1~{31}{d}/cLine1~~/rncData1~.{31}~{end}{r}
     /cMsg2~XX11~{quit}
\b   {blank XX11}~/cMsg1~XX11~{quit}
31   {end}{d 2}{end}{up}{return}
```

Msg1 Alt a   לקליטת תלמיד חדש הקש

Msg2 Alt b   הקלד את פירטי התלמיד ואח"כ

קלוט מספר תלמידים וראה כיצד כל הממוצעים משתנים בזמן אמת.

1.2.2.2 סמן לוגי - {let ... }~

לפקודת המקרו {let ... }~ יש שימושים רבים ומגוונים ולפני שניגש ללמדם ולתרגלם נלמד את מבנה הפקודה

{let y,x}~

[חייבים לסיימה בתו ~ כדי לעדכן את הגיליון].

את הפקודה הזאת נקרא כך.

הבה ו y יהיה x

let y be x

כאשר הארגומנט y הוא תמיד כתובת באחת משלושת הצורות:

1. קואורדינטות
2. שם תחום
3. פונקציה שתפוקתה כתובת

ואת הארגומנט השני אפשר לכתוב בארבעה אופנים:

1. קואורדינטות
2. שם תחום
3. פונקציה שתפוקתה כתובת
4. כל פונקציה אחרת שרוצים

בגלל המספר הרב של צירופים אפשריים אפשר להפיק מפקודה זו שימושים רבים ומגוונים. נעבור עתה על החשובים ביניהם ונתרגל אותם.

הקם סביבה לקליטת נתונים בגיליון לפי הדוגמה באיור 1.09 . שים לב שמבנה זה זהה למבנה באיור 1.05.

הקם בגיליון את כל משפטי המקרו לפי הדוגמה באיור 1.10.

הפעל את המשפטים הללו זה אחר זה לפי סדר הא"ב בשלבים וללא שלבים.

העתק לגיליון את משפטי המקרו שבאיור 1.06 ותן להם שמות חדשים.

הפעל אותם בזה אחר זה בשלבים וללא שלבים. השווה עם המשפטים מאיור 1.10.

כתוב משפטי מקרו חדשים בהם נוהל {let ... }~ מוחלף בנוהל /c.

מה ההבדלים ביניהם, היתרונות והחסרונות.

איזה נוהל אתה מעדיף ומדוע?

מתי נשתמש באחד ומתי בשני?

איור 1.09
~{let ... } במקום GoTo ובמקום Copy

	A	B	C	D	E	F	G	H
1				לוח 1.1				
2								
3	ת ק צ י ב ב ש ק ל י ם ל פ י ס ע י ף ו י ח י ד ה							
5								
5								
6								
7	XX11			ס ע י ף ת ק צ י ב י				יחידה
8	======	==	==	============	===========	====	==	===
9	5	4	3	2	1	סה"כ	שם	#
10	======	==	==	============	===========	====	==	===
11	80,442	0	0	338,534	537,326			סה"כ
12	----------	---	---		-----------------	-------		------
13				Total1				
14	3,471	0	0	5,364	22,553	31,388	א	1
15	25,671	0	0	32,495	78,483	136,649	ב	2
16	13,800	0	0	235,151	206,156	455,107	ג	3
17	37,500	0	0	65,524	230,134	333,158	ד	4
18								
19				Data1				

איור 1.10
מ ק ר ו ל א י ו ר 1.09

	J	K
1	\a	{1}XX11 ~ {let XX11,Msg1} ~ {quit}
2	\b	{1}Data1 ~ {let XX11,Msg2} ~ {quit}
3	\c	{1}Total1 ~ {let XX11,Msg3} ~ {quit}
4	\d	{1}Data1 ~ {let XX11,Msg4} ~ {end}{r}{end}{d 2}{quit}
5	\e	{1}Data1 ~ {let XX11,Msg5} ~ {end}{r}{end}{d 2}1+{u} ~ {l}{quit}
6	\f	{1}Total1 ~ {r 5}@sum({l}.{end}{l}) ~ {let XX11,Msg6} ~ {quit}
7	1	{home}{blank XX11} ~ {goto}{return}
8	Msg1	/rncXX11 תא זה הוגדר מראש ע"י
9	Msg2	Data1 הסמן הגיע לפינה השמאלית העליונה של
10	Msg3	Total1 הסמן הגיע לפינה השמאלית של
11	Msg4	הסמן התמקם לקראת הקלדה בשורה חדשה
12	Msg5	הוקלד מספר רץ לרשומה חדשה
13	Msg6	@sum כאן הוקלדה פונקצית

נבחן עתה את ההבדלים בין איורים 1.06 ו 1.10 בעזרת \b.

1. באיור 1.06 (עמוד 18)

```
\b {1}Data1 הסמן בפינת~{goto}Data1~{quit}
 1 {home}{blank XX11}~{goto}XX11~{return}
```

2. באיור 1.10 (עמוד 32)

```
\b {1}Data1~{let XX11,Msg2}~{quit}
 1 {home}{blank XX11}~{goto}{return}
```

הסמן הגיע לפינה השמאלית העליונה של Msg2 Data1

לצורך השוואה דרושה אמת מידה (מסגרת התייחסות, רשימת קריטריונים) שמשקפת למעשה נוהל חשיבה בתהליכי פיתוח. ממציא פורה לא תמיד משתמש בנוהל חשיבה ממוסד בעזרתו יניב פטנטים טובים. בתהליך היצירה שלובים גורמים רבים ביניהם מוצאים חשיבה אסוציאטיבית, שילוב כושר התמקדות עם ראיה פריפרלית (peripheral vision) לניצול דברים המזדמנים בשוליים וידע רב בתחום הנחקר ובתחומים שונים אחרים. אפשר להעיז ולהגיד שכל אותן רשימות עזר יפות וטובות שאנו מעתיקים בכיתה מהלוח הם תולדת ניתוח לאחר מעשה של מה שנראה למן דהוא כמכנה המשותף הנמוך ביותר למפתחים וממציאים רבים. אבל אין רע בכך. בעת כתיבת מקרו מומלץ לזכור ברקע את הדברים הבאים אך לתת למחשבה לרוץ באופן טבעי וללא מעצורים.

הגדרת צרכים

שני האיורים מספקים צורך כמעט זהה. רוצים לתת לתלמיד מכשיר ללימוד עצמי של אוטומציה בהולכת סמן. אביזר שיאפשר לו לראות כיצד מובל הסמן ממקום למקום ומה מתבצע משהגיע ליעד. וכל זאת בהנחה מוקדמת שהתלמיד איננו יודע דבר על אוטומציה של הולכת סמן - באיור הראשון. לעומת זאת מניחים באיור השני שהתלמיד למד ומכיר את החומר שהוצג באיור הראשון.

הגדרת מטלות

להכין ציור (נוף) עליו יובל הסמן ויתרחשו דברים.

לבחור נקודות ציון בנוף למסלול הולכת הסמן.

לבחור מקום להקרנת הסברים כל פעם שמגיעים למקום חדש.

לבחור מה לבצע בכל נ.צ. שמגיעים אליו.

לכתוב את הטקסטים שיוקרנו כדיברי הסבר.

נסה להוסיף עוד מטלות לרשימה זו ע"י התבוננות באיורים.

עדיין יש זהות מלאה בהגדרת המטלות בין שני האיורים הללו.

בחירת שיטה

כאן לראשונה יש ביטוי להבדלים בין שני האיורים.

השיטה באיור הראשון מבוססת על נוהל הולכת סמן בפועל:

1. הולכת סמן ע"י תחליף למקשי החיצים למקום בו יופיעו הסברים,
2. הקלדה ע"י מקרו כאשר ההסבר הוא חלק ממחרוזת המשפט הראשי,
3. הולכת הסמן אל נ.צ. בנוף,
4. ביצוע (או אי ביצוע) שינויים בנוף במקום החדש,
5. חשיפה ראשונה לעקרון המודולריות.

השיטה באיור השני מבוססת על נוהל הולכת סמן לוגי ב ~{ ... let}:

1. יישום עקרון המודולריות לשילוב טקסטים במקרו בנוהל Msg1,
2. הקרנת ההסבר מבלי להזיז את הסמן ממקומו,
3. הולכת הסמן אל נ.צ. בנוף,
4. ביצוע (או אי ביצוע) שינויים בנוף במקום החדש,
5. הצגת שימושים נוספים של עקרון המודולריות.

נסה להוסיף לרשימה זו ע"י התבוננות באיורים.

לאחר כל אלה כתוב את סדר הפעולות המפורט של משפט \b בשני האיורים והשווה ביניהם. אח"כ הפעל אותם שוב בשלבים.

למדנו לשכפל תוכן טקסט או מספר מכתובת אחת לשניה כאשר הסמן נמצא אי שם בגיליון ולא זז ממקומו בעת ביצוע השיכפול. נלמד עתה לשלב משוואות ב ~{ ... let} ולהפעילן ממשפט המקרו במקום לכתוב משוואה חיה בגיליון.

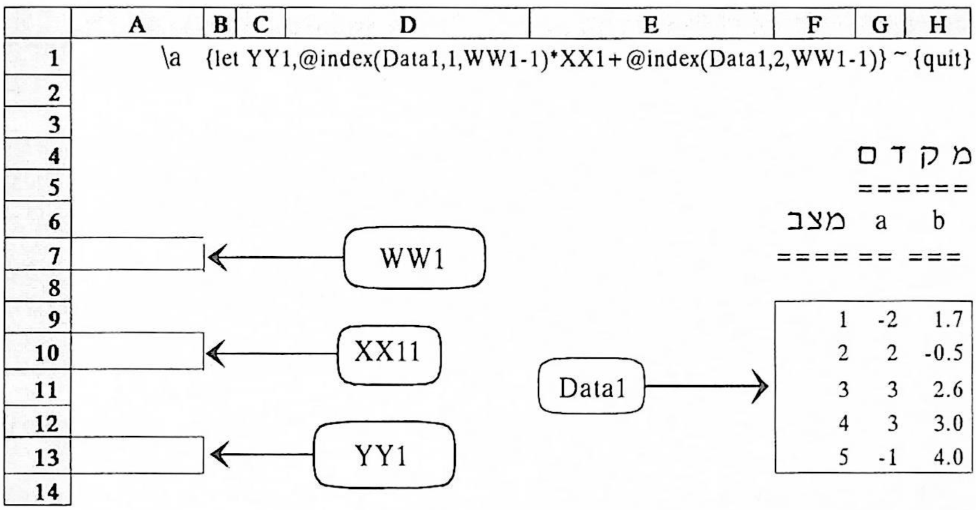
חוקרים, מדענים, מהנדסים, טכנאים ובעלי מקצוע שונים זקוקים לעיתים מזומנות למספרים שנשלפים מלוחות. יש שנושאים בתיקיהם ספרים עבי כרס ויש גם כבר לוחות שמוחשבו. אבל זו עדיין מטלה מתישה. כאן אפשר להכניס לפעולה את האוטומציה בגיליון על כל מרכיביה. נראה דוגמה פשוטה.

נאמר שהמשוואה הבאה מתארת קשר בין שני משתנים

$y = ax + b$

נאמר גם שבמצבים שונים יש ל a ול b ערכים שונים ואלה נמדדו בסידרת ניסויים ונרשמו בלוח הבא.

תרגיל 1.09
שליפת נתונים ע"י שילוב משוואה ב ~{let ... }



בצע את התרגילים הבאים.

הקם במחשב שלך את הגיליון שבתרגיל 1.09.

מצא היכן ב \a כתובה המשוואה

$y = ax + b$

מה פשר הכתובות

WW1 XX1 YY1

ומדוע כפל האותיות?

מדוע WW1 ולא AA1 ?

הכנס מספרים בשניים משלושת התחומים הללו:

1. מדוע רק בשניים?
2. באיזה שניים מתוך השלושה?
3. מה הם ערכי המינימום והמקסימום בכל אחד משניהם?

חייבים לסיים ב ~{let a,b} ב ~ כדי לעדכן את הגיליון

לצד הימני של המשוואה שני מרכיבים ושניהם כתובים בארגומנט השני של ה ~{let ... } ב \a . כתוב כל אחד בנפרד בגיליון כפונקציה חיה ולא כטקסט. שנה את המספרים ב WW1 וב XX1 וראה כיצד תפוקת הפונקציות משתנה. נסה עם מספרים מעבר למקסימום ולמינימום.

מדוע WW1-1 ?

הפעל את \a בשלבים ולא בשלבים, עם פרמטרים נכונים ועם ערכים מעבר למקסימום ולמינימום.

החלף את הפונקציה Index בפונקציה LookUp וחזור על הניסוי. מה ההבדלים בין שתי הפונקציות ומתי נשתמש בכל אחת מהן?

אחרי מספר ניסויים עולה תחושה של חוסר מסויים. הרגשה שהקמנו קטע אחד בלבד מתוך שרשרת מטלות שאפשר לגשר ביניהן באופן אוטומטי ואינטראקטיבי.

שליפת פרמטרים מלוח היא רק חוליה בשרשרת של מטלות. רשום מטלות שכדאי לבצע לפני ואחרי שליפת הפרמטרים.

נעבור עתה למטלה נוספת לביצוע ב ~{let ... }.

נוסיף בגיליון התרגיל הנוכחי את המקרו הבא.

```
\b {let WW1,WW1-1}~
   {let  YY1,@index(Data1,1,WW1)*XX1+@index(Data1,2,WW1)}~{quit}
```

חזור עם משפט זה על סידרת הניסויים שערכת זה עתה.

השווה בין שני המשפטים הללו ורשום את משמעות ההבדל.

איזה רעיונות יש לך עכשיו, אחרי שראית שימוש אחד בנוהל

{let WW1,WW1-1}~

1.2.2.3 סמן לוגי - { i n d i c a t e ... }

תרגיל 1.10
ד ו - ש י ח מ ע ל ה ס ר ג ל ה ע ל י ו ן

A1:		שורה 1	איזור הדו-שיח:	READY
		שורה 2		
		שורה 3		

	A	B C D E F G H
1	\a	{indicate Msg1}{quit}
2	\b	{indicate}{quit}
3	Msg1	אפשר לכתוב עד 240 תווים אבל על המסך נראה 80 בלבד
4		

הגיליון האלקטרוני מרכיב רשת קואורדינטות על רוב שטח המסך ומשאיר את שלושת השורות שמעל הסרגל העליון לניהול דו-שיח בין המחשב והמשתמש. אבל הגיליון ה"חכם" נותן אופציה למשתמש להפעיל גם חלק זה כרצונו. ומאחר שללא קואורדינטות אין סמן בפועל צריך להפעיל סמן לוגי באמצעות פקודות מקרו. יש מספר פקודות כאלה למטרות שונות. הפקודה שנלמד כאן נועדה להקרין טקסט בשורה הראשונה מלמעלה מבין שלושת השורות שמעל הסרגל העליון.

פקודת המקרו היא

{indicate a}

בארגומנט אפשר לכתוב טקסט בגרשיים

{indicate "המתן"}

או להשתמש בנוהל Msg1

{indicate Msg1}

בתחום Msg1 כותבים טקסט או נוסחאות שמניבות טקסט או שילוב של טקסט עם משוואות. לנוהל זה יתרונות על פני כתיבת הטקסט בתוך הארגומנט - שילוב טקס אחד במספר מקומות במשפט המקרו מבלי שיהיה צורך לכתוב אותו יותר מפעם אחת, אפשרות ביצוע שינויים בטקסט בלי לחפשו והחדרתו בדרך זו אל המשפטים הראשיים בכל מקום שמתבקש.

אין צורך לסיים את הטקסט של Msg1 בפקודה {return} מכיוון שהוא כתוב כארגומנט ולא כקריאה למשפט שירות.

הפעלת פקודה זו כחלק מרצף תווים במשפט מקרו מקרינה את הטקסט

בשורה הראשונה של איזור הדו-שיח. כדי לחזור למצב READY כותבים, ללא ארגומנט, את הפקודה

{indicate}

העתק לגיליון את תרגיל 1.10 והפעל את המקרו בשלבים וללא שלבים. כתוב טקסטים שונים ב Msg1 והפעל.

שלב ב Msg1 נוסחה שמניבה טקסט (פונקצית (@string(...) להחדרת משתנה דינמי אל תוך ההודעה המוקרנת.

במהלך הספר תופעל פקודה זו מספר פעמים. דוגמאות נוספות נראה בתרגיל הבא.

תרגיל 1.11
נוהל {indicate MsgX}

	A	B	C	D	E	F	G	H
1	\a {indicate @@("Msg"&sNNx)}{quit}							
2	\b {indicate}{quit}							
3	NNx	1						
4	sNNx	1	< --- @string(NNx,0)					
5	Msg1	המתן						
6	Msg2	ה מ ת ן						
7	Msg3		< --- @repeat(" ",80)					
8	Msg4		< --- @repeat(". . . . : . . . . ^",8)					
9	Msg5	P r i n t e r e r r o r. . . ENTER בדוק מדפסת ואח"כ						
10								

בדו-שיח "אמיתי" עם המחשב רוצים לקבל ממנו הודעות על מה שמתרחש בגיליון המופעל ע"י מקרו, תוך כדי ההתרחשות. לביצוע מטלה זו מדגים תרגיל 1.11 כיצד אפשר להחדיר אפקט של שינויים בגיליון אל תפוקת {indicate}. נסביר תחילה את המבנה ואת הפונקיות ואח"כ נתרגל.

@@(Adr1) ---------> ה מ ת ן

Adr1 | Adr2 |

Adr2 | ה מ ת ן |

התו @ נקרא באנגלית - At. ולכן נקרא לפונקציה שלמעלה - At At.

פונקציה זו פועלת כך - "גש אל חדר א. על השולחן תמצא פתק ובו כתוב כיצד להמשיך". נגשנו לחדר א, מצאנו את הפתק ובו היה כתוב - "גש לחדר ב. על השולחן יש פתק ובו כתוב מה צריך לעשות". נגשנו לחדר ב ובפתק שמצאנו כתוב לאמור - "בדוק מדפסת ואח"כ ENTER . . . Printer Error".

ובדוגמה שלפנינו - אם בכתובת Adr2 כתוב "המתן" ובכתובת Adr1 כתוב Adr2 אזי תפוקת הפונקציה @@(Adr1) היא המלה "המתן".

NNx	1
sNNx	1

< --- @string(NNx,0)

הפונקציה (...)@string הופכת VALUE ל LABEL. הארגומנט הראשון מצביע על הכתובת שם נמצא המספר שרוצים לקבלו כטקסט. הארגומנט השני מגדיר כמה ספרות עשרוניות רוצים שיופיעו בתפוקת הפונקציה.

ערוך מספר נסיונות בגיליון. אם למספר יש 3 ספרות עשרוניות והארגומנט השני מגדיר רק 2 - מה נקבל? האם הפונקציה מעגלת את הספרה השניה או פשוט משמיטה את השלישית?

בדוגמה הנוכחית, כל מספר שיכתב ב NNx יופיע בזמן אמת כטקסט ב sNNx.

+"Msg"&sNNx

לפנינו משוואה שמחברת שני טקסטים (concatenation) כאשר התו המחבר איננו הסימן + אלא הסימן & (ampersand). הטקסט הראשון נתון בגרשיים כפולים - "Msg" מכיוון שהוא כתוב כחלק של המשוואה. החלק השני הוא שם תחום שמכיל טקסט ולכן אין צורך להקיפו בגרשיים. שני החלקים ביחד מפיקים שם תחום המורכב מטקסט קבוע - Msg וממספר משתנה. אם נכתוב ב NNx מספר מסוים נקבל אותו מיד כטקסט ב sNNx והמשוואה לחיבור טקסטים תפיק שם של תחום. הלוח הבא מראה את תפוקת המשוואה כתוצר של שינויים ב NNx.

אם	אז
1 = NNx	+"Msg"&sNNx = Msg1
2	Msg2
3	Msg3
4	Msg4
5	Msg5

כאשר נכתוב את המשוואה

@@("Msg"&sNNx)

אז אם ב NNx כתוב 1 נקבל את התוכן של Msg1 כלומר - המתן. וכך הלאה. ומכיוון שמשוואה זו משולבת במקרו

{indicate @@("Msg"&sNNx)}

כל פעם שיופעל משפט זה תופיעה בשורת הדו-שיח הראשונה ההודעה שכתובה ב Msg שמספרו כמספר שב NNx.

העתק את חיבור המקרו מתרגיל 1.11 והקם את כל התחומים.

בצע ללא שלבים את הפעולות הבאות

הקש 1 ב NNx ואחר כך Alt a. החלף את המספרים ב NNx והפעל שוב את המקרו אחרי כל החלפה.

רשום את סדר הפעולות המפורט של חיבור זה כאשר NNx = 1.

יש דרכים שונות לקבל אפקט דומה. דוגמה זו נבחרה כדי להציג פונקציות נוספות וחשובות וכדי להראות העמקה נוספת של כתיבה מודולרית ושל שתילת ארגומנט דינמי בתוך משפט מקרו.

1.2.2.4 סמן לוגי - ~{let W,@index(Data1,X,Y)}

שילוב פונקצית האינדקס עם פקודת המקרו ~{let ... } מניב שליפת נתון מבסיס נתונים וכתיבתו במקום כל שהוא בגיליון בלי להזיז את הסמן ממקומו, בלי לכלול משוואות חיות בגיליון, בזמן אמת ובאופן המאפשר הפעלה בשרשרת משפטים. את מבנה פונקצית האינדקס למדנו בעמודים 9..12 ואת הפקודה ~{let ... } בעמודים 31..37. כאן, בעזרת תרגיל 1.12 נלמד לשלבם.

העתק את תרגיל 1.12 לגיליון, ערוך מספר ניסויים וענה על השאלות.

הכנס מספרים שונים ב MM1 וב NN1 והפעל את משפט המקרו. ערוך מספר ניסויים.

מדוע MM1-1 ו NN1-1?

רשום את סדר הפעולות המפורט של המשפט \a.

תרגיל 1.12

{let W,@index(Data1,X,Y)} ~

	A	B	C	D	E	F	G
1	\a {let Print1,@index(Data1,MM1-1,NN1-1)} ~ {quit}						
2	MM1	☐	<--- מספר הסעיף התקציבי				
3	NN1	☐	<--- מספר המחלקה				
4							
5							
6		Print1		☐			
7							
8				לוח 1.12			
9				תקציבי מחלקות לפי סעיפים			
10							
11				סעיף תקציבי		מחלקה	
12				============	============	====	======
13				1	2	#	שם
14				============	============	===	======
15							
16			0	25	500	1	בגדים
17			1	15	350	2	וידיאו
18	Data1 ——>		2	12	275	3	ספרים
19			3	35	750	4	ספורט
20				0	1	2	3

1.2.2.5 סמן לוגי - {let W,@vlookup(X,Data2,Y)} ~

בדוחות רבים אנו רוצים להציב את שם החודש בעברית במקום באנגלית. נלמד כאן כיצד לבצע את המטלה הזאת באופן אוטומטי.

לפנינו ארבע פונקציות תאריך והתפוקות שלהן. הראשונה מפיקה את התאריך הנוכחי, השניה את היום בתאריך הנוכחי, השלישית את החודש והרביעית את השנה הנוכחית. מה שאנחנו למעשה רוצים זה שבמקום המספר 12 נקבל את המלה דצמבר. לשם כך נכין תשתית פשוטה.

```
@now            ->    25.12.91
@day(@now)      ->    25
@month(@now)    ->    12
@year(@now)     ->    91
```

Data2 →

1	ינואר
2	פברואר
3	מרץ
4	אפריל
5	מאי
6	יוני
7	יולי
8	אוגוסט
9	ספטמבר
10	אוקטובר
11	נובמבר
12	דצמבר
0	1

בעמודה הראשונה (מספר 0) של התחום Data2 רשומים הערכים - VALUE - 12 .. 1 עבור כל אחד מחודשי השנה ובעמודה השניה (מספר 1) רשומים שמות החודשים בעברית. המשוואה הבאה מפיקה את שם החודש בעברית.

@vlookup(5,Data2,1)	->	מאי
@vlookup(8,Data2,1)	->	אוגוסט
@vlookup(@month(@now),Data2,1)	->	דצמבר

(אם דצמבר הוא החודש הנוכחי)

שלא כפונקצית האינדקס אין פונקציה זו שולפת על פי כתובת של שורה ועמודה במטריצת הנתונים. החיפוש כאן מתבצע תמיד לאורך העמודה הראשונה (מספר 0) מלמעלה למטה כאשר המחשב מחפש בעמודה זו את מה שכתבנו בארגומנט הראשון (X) של הפונקציה - למשל את המספר 5 . כאשר נמצא המספר (או הטקסט) המבוקש נפסק החיפוש והמשוואה מניבה את מה שכתוב באותה שורה ובעמודה הרשומה בארגומנט השלישי (Y) - עמודה מספר 1 בדוגמה הנוכחית. תחום הנתונים מוגדר בארגומנט השני - Data2.

להפעלה נכונה של פונקצית החיפוש האנכי חייבת העמודה הראשונה להיות ממוינת לפי סדר עולה. הבה נבדוק מה יקרה אם המיון לא יהיה בסדר עולה.

בלוח הבא השתבש סדר החודשים וחודש יוני מופיע לפני חודש מאי. פונקצית החיפוש האנכי פועלת בשלבים. החיפוש (למשל אחרי הערך 5) מתחיל בשורה הראשונה של העמודה הראשונה. כאן מתעכב המחשב ושואל: "האם הערך כאן שווה 5 או גדול מ 5 ?". כמובן שהמספר 1 מוליד את התשובה "לא" והמחשב יורד תא אחד בעמודה זו

	ינואר	1	↓
	פברואר	2	צעד 1
	מרץ	3	
***	אפריל	4	צעד 2 ↑
**	יוני	6	↓ ↑
*	מאי	5	
	יולי	7	
	אוגוסט	8	
	ספטמבר	9	
	אוקטובר	10	
	נובמבר	11	
	דצמבר	12	

וחוזר ושואל אותה שאלה. וכך הוא ממשיך עד שהגיע לשורה של חודש יוני. כאן רשום המספר 6 והוא גדול מהמספר 5 . לפיכך התשובה למחשב היא "כן" ומסתיים הצעד הראשון בחיפוש.

אילו היה כתוב 5 היה המחשב שולף את המלה "מאי" אבל הרי כתוב 6 והוא גדול מ 5 . במקרה זה עוצר המחשב בתא שבו כתוב 6 וחוזר תא אחד למעלה (שכתוב בו 4) ושולף "אפריל".

במילים אחרות. התניית החיפוש היא: אם שווה או גדול, עצור, אם שווה - שלוף מהשורה הנוכחית אחרת חזור שורה אחת ושלוף ממנה. לכן הפעלת פונקצית החיפוש מחייבת בסיס נתונים ממוין.

נחזור לענייננו. את התאריך הנוכחי רוצים לקבל בצורה הבאה:

ד צ מ ב ר 2 5 , 9 1

כלומר כמחרוזת טקסט. נהפוך תחילה כל אחת מפונקציות התאריך לטקסט ולאחר מכן נשלב אותן במחרוזת אחת.

1. ערכים (VALUE) ומחרוזות (LABEL)

```
A      @now                          ->   25.12.91
       @day(@now)                    ->   25
       @vlookup(@month(@now),Data2,1) ->  דצמבר
       @year(@now)                   ->   91
```

זהה בין ארבעת הפונקציות הללו את השלוש שמניבות ערכים ואת האחת שמפיקה מחרוזת.

2. פונקציות מחרוזת

B

```
@string(@day(@now),0)                ->    25
@vlookup(@month(@now),Data2,1)       ->    דצמבר
@string(@year(@now),0)               ->    91
```

3. שילוב מחרוזות ותוספות "קוסמטיות".

שילוב הפונקציות:

C

```
@string(@year(@now),0)&","&@string(@day(@now),0)&"  "&@vlookup(@month(@now),Data2,1)
```

מ ח ז י ר : דצמבר 25, 91

נניח שלפני שמתחילים להדפיס דוח רוצים להציב את התאריך הזה במקום ששמו Date1. המקרו הבא יבצע זאת במהירות ובלי שיהיו פונקציות חיות בגיליון.

```
{let Date1,@string(@year(@now),0)&","&@string(@day(@now),0)&"  "&@vlookup(@month(@now),Data2,1)}~
```

הקם את בסיס הנתונים לחודשים והעתק את הנוסחאות ב A ב B וב C לגיליון. ערוך ניסויים שונים ואח"כ כתוב את המקרו והפעל אותו.

1.2.2.5 סמן לוגי - ~ {put ...}

למדנו להשתמש בפונקצית אינדקס לשליפת נתונים מלוח על ידי הגדרת כתובת במטריצה בשיטת הקואורדינטות. האם יש נוהל לביצוע תהליך הפוך דהיינו, החדרת נתון אל כתובת בתוך לוח עם נתונים על ידי קואורדינטות הכתובת הזאת בלוח.

קיים הבדל מענין בין פונקציות לבין מקרו. כל פונקציה יכולה להתייחס - על פי הגדרותיה - לכתובות שונות בגיליון אך תפוקתה תמיד מופיעה אך ורק בתא בו היא כתובה. לפיכך יש פונקציות לשליפת נתונים אך אין פונקציות להחדרת נתונים אל כתובת שונה מכתובתה הנוכחית של הפונקציה. לעומת זאת מקרו יכול להעביר נתונים מכל כתובת אל כל כתובת בגיליון. הפקודה ~{let ...} מעבירה נתונים מתא אחד לשני וכשמשלבים בה פונקצית אינדקס היא יכולה להוליך סמן לוגי על פני רשת קואורדינטות במטריצה ממנה נשלפים הנתונים. לפקודה ~{put ...} יכולת נוספת - הולכת סמן לוגי על פני רשת קואורדינטות במטריצה לתוכה שולחים את הנתון.

במילים אחרות, פונקצית אינדקס שולפת נתון מתא באיזור מוצא ומקרו ~{put ...} כותב נתון בתא באיזור יעד. שילוב שני

אלה מעתיק נתון מתא באיזור מוצא אל תא באיזור יעד.

מרכיבי הפקודה יראו מוכרים מכיוון שההגיון שעמד לנו עד כה ממשיך להנחותנו גם כאן ובהמשך.

{put Data1,X,Y,Z} ~

ומשנפרק פקודה זו לגורמים נקבל:

{put	מלת הפעלה - שים
Data1	שם התחום בו יקלט הנתון
X	מספר העמודה ב Data1 כאשר X = x .. 0
Y	מספר השורה ב Data1 כאשר Y = y .. 0
Z}	כתובת המקום בו נמצא הנתון שישלח אל Data1
~	ENTER

תרגיל 1.13

{put Data1,X,Y,Z} ~

	A	B	C	D	E	F	G
1	\a {put Data1,MM1-1,NN1-1,Read1} ~ {quit}						
2	MM1	☐	<--- מספר הסעיף התקציבי				
3	NN1	☐	<--- מספר המחלקה				
4	Read1	☐	<--- מה שרוצים לכתוב בלוח				
5							
6							
7							
8					לוח 1.13		
9				תקציבי מחלקות לפי סעיפים			
10							
11				סעיף תקציבי			מחלקה
12				==========	==========	====	======
13				1	2	#	שם
14				==========	==========	===	======
15							
16			0	25	500	1	בגדים
17			1	15	350	2	וידיאו
18	Data1 ⟶		2	12	275	3	ספרים
19			3	35	750	4	ספורט
20				0	1	2	3

העתק את התרגיל לגיליון, ערוך מספר ניסויים וענה על השאלות.

הכנס מספרים שונים ב MM1 וב NN1 והפעל את משפט המקרו. ערוך מספר ניסויים.

מה יעודו של התחום Read1. הכנס בו מספרים שונים (או טקסט) והפעל את משפט המקרו עם ערכים שונים ב MM1 וב NN1.

מדוע MM1-1 ו NN1-1?

רשום את סדר הפעולות המפורט של המשפט a\.

מה זהה ומה שונה בין המשפט בתרגיל 1.12 לבין המשפט בתרגיל 1.13.

מאחר וספר זה נועד ללימוד עצמי ומטרתו לעזור בפיתוח כושר ניתוח וחשיבה עצמי נציג עכשיו בעיה ונבקש ממך לפתור אותה. הפתרון ינתן בהמשך כדי שתוכל להשוות ולענות על שאלות נוספות.

יש שתי מטריצות עם נתונים. באחת - Data1 - יש שתי עמודות וארבע שורות. בשניה - Data2 - יש שלוש עמודות ושתי שורות.

רוצים להעתיק נתון שנמצא ב Data1,MM1,NN1 אל מקום שנמצא בכתובת Data2,MM2,NN2.

מטלה זו תתבצע על ידי פקודת מקרו ~{put ... } אחת.

הקם את המבנה בגיליון, הגדר תחומים, כתוב את המקרו והפעילו.

כתוב את סדר הפעולות של המקרו שכתבת.

השווה את הפיתרון שלך עם הפיתרון בתרגיל 1.14.

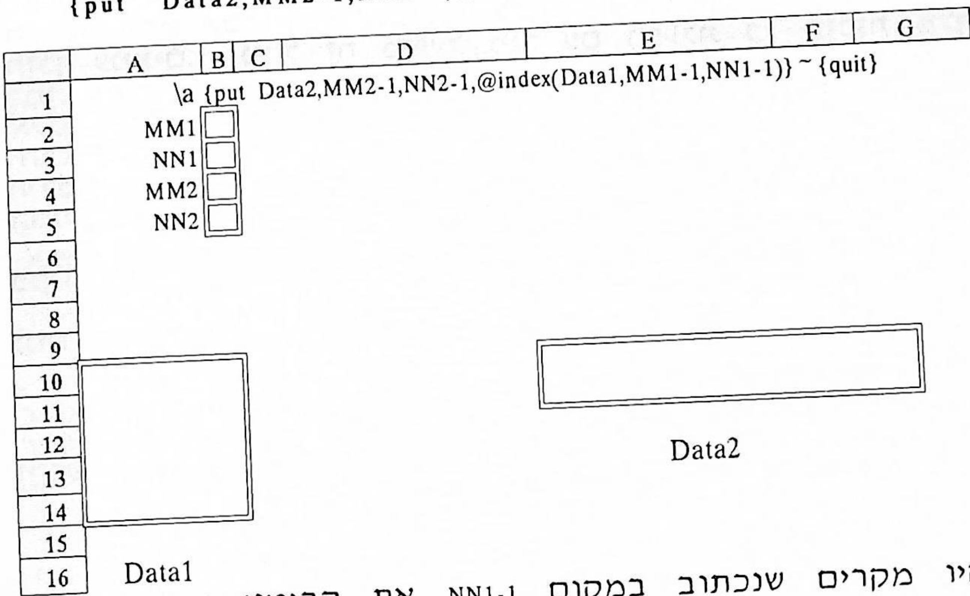
הפעל את המקרו עם ערכים שונים לארגומנטים של הקואורדינטות.

כתוב את סדר הפעולות של משפט המקרו בתרגיל 1.14.

מקרו ~{put ... } חייבים לסיים בסימן ~ כדי לעדכן את הגיליון.

תרגיל 1.14

{put Data2,MM2-1,NN2-1,@index(Data1,MM1-1,NN1-1)} ~



יהיו מקרים שנכתוב במקום NN1-1 את הביטוי NN1-X כאשר X = פונקציה כל שהיא או מספר כל שהוא. פעולה זו מתרחשת כאשר רוצים שתנועת הסמן לא תתבצע בעמודות מסוימות. תכנן ובצע תרגיל לדוגמה. על איזה עמודות לא יוכל הסמן הלוגי לנוע?

שלושת הארגומנטים הראשונים של המקרו נועדו לזיהוי כתובת היעד של הנתון המועתק. הארגומנט הרביעי נועד להצביע על כתובת המוצא ממנה יועתק הנתון. ארגומנט רביעי זה, כשהוא כתוב כפונקצית אינדקס מאפשר להעתיק מכתובת במטריצה אחת אל כתובת במטריצה אחרת. טכניקה זו חיונית ליישומים מתקדמים שנלמד בהמשך ספר זה ובספרים הבאים בסידרה זו.

1.3 מקרו אינטראקטיבי

מטלות ממוחשבות רבות נדרשות למידע מהאדם תוך כדי הפעילות האוטומטית של המחשב. תוך כדי ביצוע המקרו נזקק המחשב למידע כל שהוא ולפיכך עלינו לגרום לו שיעצור לרגע, ימתין עד שנקליד נתונים כל שהם ואח"כ ימשיך לפעול תוך כדי התייחסות אל המידע החדש שהגיע מאיתנו. למשל, אם מפעילים מקרו לשליפת כרטיס של פריט מלאי מסוים צריך לשלב פקודת מקרו במשפט אשר תעצור לרגע את ריצת המחשב ותאפשר המשכת הפעילות האוטומטית אחרי שהקלדנו את שמו או את מספרו של הפריט המבוקש. או, לחילופין, יעצר המקרו עד שנציב את הסמן על הפריט המבוקש ונקיש ENTER.

1.3.1 מקרו אינטראקטיבי - {?}

השימוש הקלאסי והנפוץ ביותר במקרו אינטראקטיבי הוא נוהל תוכן עניינים. נוהל זה מופעל מיד עם הפעלת כל תוכנת מדף כמו משחקי מחשב, מעבדי תמלילים, תוכנות גראפיות, גליונות אלקטרוניים, הנהלת חשבונות וכל מה שתרצו. במהלך ספר זה, והבאים אחריו, נתייחס לנושא זה ברמות שונות ובהתאמה לענין הנלמד באותו זמן. כאן נלמד את המרכיבים הבסיסיים ואת התשתית ההגיונית הפשוטה. אך תחילה ל"מעצור" האינטראקטיבי.

פקודת המקרו {?} כשהיא משולבת במשפט מקרו פועלת כך:

נניח שכתבנו

```
\z {goto}{?}~{quit}
```

והפעלנו את המקרו. אזי

1. מפעילים את המקרו	Alt z
2. מקרו במקום המקש GoTo (F5)	{goto}
3. המקרו עוצר ורואים מעל הסרגל העליון את הדרישה - Enter address to go to:	{?}
4. מקלידים את כתובת היעד - b11	b11
5. מקישים על המקש ENTER	ENTER
6. המחשב קולט את כתובת היעד - b11	
7. תחליף מקרו למקש ENTER	~
8. סוף	{quit}

כתוב את המקרו הזה בגיליון והפעל אותו מספר פעמים.

נבחן עתה מקרוב את מהותו וסדר פעולותיו של נוהל תוכן העניינים. נפנה את מחשבתנו לתוכן עניינים רגיל בספר. מעוניינים לקרוא פרק מסויים ורוצים לדעת את כתובתו בספר. פותחים את דף תוכן העניינים - שתמיד ממוקם בתחילת הספר, בעזרת העין ואצבע מוצאים את הפרק המבוקש ולידו הכתובת - מספר הדף בו מתחיל הפרק בספר, ומדפדפים עד שמגיעים למספר הנכסף.

ההבדל בין נוהל תוכן עניינים ידני לבין הממוחשב גדול באפקט אך לא שונה במהות. סדר המטלות יהיה כך:

1. הפעלת מקרו להקרנת תוכן עניינים על המסך
2. עצירת המקרו תוך כדי הפעלתו ומבלי להפסיקו
3. מתן אפשרות להוליך את הסמן כדי להצביע על הפרק המבוקש

או להקליד את מספר הפרק המבוקש
4. עם הקשת ENTER יוצב הדף הראשון של הפרק המבוקש על המסך

סדר מטלות זה (להבדיל מסדר פעולות) רומז על פעולה אינטראקטיבית נוספת. ננסח זאת על ידי שאלה:

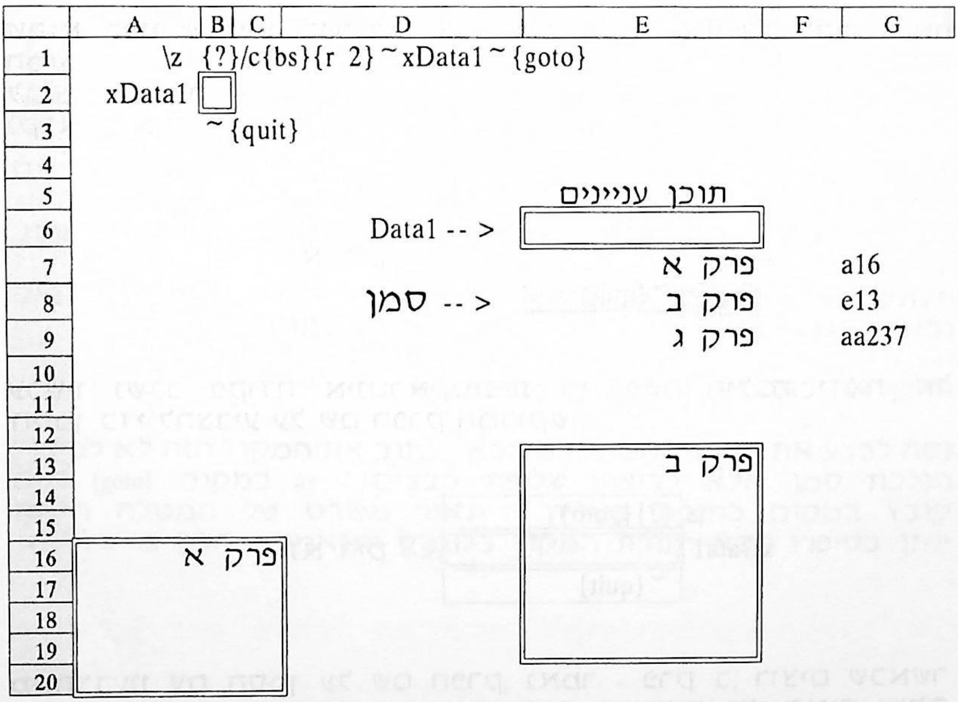
הצבענו ע"י הסמן על הפרק המבוקש והקשנו ENTER

היכן נרשום את כתובת הפרק וכיצד נחדיר אותה אל תוך משפט המקרו?

1.3.2 מקרו אינטראקטיבי - תוכן עניינים להולכת סמן בגיליון

נחזור לספר ולפעולה הידנית. האצבע מצביעה על שם הפרק ואח"כ נעה ימינה ומצביעה על כתובת הפרק בגיליון. המוח קולט וזוכר את הכתובת עד שנמצא אותה. אוטומציה זה, לכל דבר ועניין, שיחזור של פעולה ידנית. נעשה זאת בעזרת האיור הבא.

איור 1.11
מבנה משפט מקרו לשילוב ארגומנט דינמי



באיור 1.11 מודגם העיקרון הבסיסי על חלקיו השונים. במרכז המסך רואים את תוכן העניינים עצמו המורכב מכותרת, שורת רווח ושמות הפרקים. שורת הרווח הוגדרה בשם התחום Data1. כל אלה נמצאים בעמודה e משורה 5 ומטה. העמודה שמימין f היא עמודת רווח ומימין לה, בעמודה g כתובתו של כל פרק.

פקודת המקרו הטבעית להולכת הסמן אל כתובתו של פרק היא {goto}. אולם הארגומנט (כתובת היעד) לפקודה זו נמצא מחוץ לסוגריים המסולסלים. בעמוד 16 למדנו דרך אחת לשלב ארגומנט דינמי עם פקודה זו ועכשיו נלמד שיטה נוספת. הבה נכתוב משפט מקרו שמילותיו כתובות אחת מתחת לשניה בעמודה אחת במקום זו ליד זו בתא אחד.

\z	{goto}	
		<-- תא ריק
	~{quit}	

עם הפעלתו יקרא המחשב את המשפט הזה מלמעלה למטה ויבצע את הוראותיו. במהלך המקרו אפשר לכתוב בתא הריק כתובת כל שהיא לפני שהגענו לפקודה {goto} ומשימשיך המחשב לקרוא את המשפט תהיה כבר כתובת היעד משולבת בו. יש דרכים שונות לביצוע מטלה זו וכאן נעשה זאת על ידי פקודת c/. תחילה נקרא לתא הריק שבתוך המקרו בשם, נאמר xData1.

\z	{goto}	
xData1		<-- תא ריק
	~{quit}	

עכשיו נשלב פקודה אינטראקטיבית שתאפשר הולכה ידנית של הסמן כדי להצביע על שם הפרק המבוקש.

\z	{?}{goto}	
xData1		<-- תא ריק
	~{quit}	

משהצבענו עם הסמן על שם הפרק, נאמר - פרק ב, רוצים שכאשר נקיש ENTER יעתיק המחשב את הכתובת שנמצאת שני תאים ימינה

ממנו אל התא הריק שבתוך משפט המקרו וימשיך בביצוע.

\z	{?}/c{bs}{r 2}~xData1~{goto}	
xData1		<-- תא ריק
	~{quit}	

עתה משנפעיל את המקרו יאפשר סימן השאלה להניע את הסמן ולהצביע על "פרק ב". כשנקיש ENTER יעתיק המחשב את כתובת היעד אל התא הריק ויוליך אותנו לשם.

\z	{?}/c{bs}{r 2}~xData1~{goto}	
xData1	e13	התא הריק
	~{quit}	קיבל נתון

a16	פרק א
e13	פרק ב
aa237	פרק ג

כך "לומד" המחשב כל פעם כתובת של פרק אחר לפי רצוננו.

הערה. שם המקרו \z איננו מקרי. מכיוון שתוכן עניינים מופעל לעיתים קרובות נהוג להשתמש בצמד המקשים Alt z להפעלתו מפני שהם קרובים מאד וקל להקישם באצבעות יד אחת.

העתק את המבנה שבאיור 1.11 לגיליון והפעל אותו בשלבים ובאופן רגיל. בצע זאת מספר פעמים בצורות שונות.

החלף את כתובות הפרקים בשמות תחום.

נסה לבצע את התרגיל המתקדם הבא. כתוב את המקרו הזה לא לביצוע הולכת סמן אלא לביצוע שליפת קבצים - /fr במקום {goto} ושם קובץ במקום כתובת בגיליון. תאור מפורט של המטלה הזאת ינתן בסיפרי הבא: "תורת המקרו בלוטוס ותואמיו - חלק ב: שליטה".

1.3.3 מקרו אינטראקטיבי - ~{getnumber/getlabel}

בעמוד 36 בסיומו של תרגיל 1.09 נאמר:

"שליפת פרמטרים מלוח היא רק חוליה בשרשרת של מטלות. רשום מטלות שכדאי לבצע לפני שליפת הפרמטרים..."

בתרגיל 1.12 (עמוד 41) למדנו לשלוף נתון מלוח אחרי שהקלדנו שני נתונים - מספר הסעיף התקציבי ומספר המחלקה ובתרגיל 1.14 (עמוד 47) למדנו להעתיק נתון מכתובת בלוח אחד אל כתובת בלוח שני.

עבור שוב על כל התרגילים הללו ורענן את זיכרונך. אחר כך נסה לענות בכוחות עצמך על השאלה הבאה לפני שתמשיך לקרוא.

לכל שלושת התרגילים הללו יש מטלה משותפת. מה היא מטלה זו?

בכל אחת מהדוגמאות הללו חייבים להזין את המחשב במספר נתונים לפני שאפשר יהיה להפעיל את משפט המקרו ולקבל ממנו תפוקה שיש לה משמעות לגבינו. בכל אחת מהדוגמאות הללו ביצענו את המטלה הזאת באופן ידני. עכשיו נלמד לבצע הקלדת פרמטרים בנוהל אינטראקטיבי במהלך פעולתו של משפט מקרו.

אחת התפוקות החשובות מפיתוח כושר לימוד עצמי היא היכולת לפתור באופן עצמאי לגמרי בעיה מזדמנת. סוג כזה של בעיה מופיע כל אימת שאנו מעדכנים (up grading) את גירסת הגיליון שלנו. קפיצה גדולה וחשובה באוטומציה בגיליונות אלקטרוניים התרחשה כאשר חברת לוטוס יצאה עם גירסאות 2 ומעלה. אחרי גירסה 2.01 נוספו מפעם לפעם מספר פקודות מקרו והנסיון מראה שזה גם מה שצפוי בעתיד. מי שפיתח דרך חשיבה ונסיון באוטומציה לא יתקשה ללמוד בעצמו את התוספות שבגירסאות החדשות. אדרבה, הנסיון מראה שהאתגר הזה גורם הנאה וסיפוק. בפרט שהדברים הנוספים בדרך כלל עונים על נושאים שגם אנחנו חיפשנו להם פיתרון קל.

לפיכך, למי שיכול ורוצה, ממומלץ לפתור את תרגיל 1.15 בדרך הבאה.

1. העתק את התרגיל במלואו אל הגיליון, הגדר את התחומים והקם את משפטי המקרו. במילים אחרות:

   הקם את מערך האינטרה מקרו שבתרגיל 1.15.

2. הפעל אותו בשלבים ובאופן רגיל.

3. ערוך ניסויים שונים ונסה להבין בכוחות עצמך כל מה שאפשר ללמוד מחיבור האינטרה מקרו הזה.
4. השווה את מה שלמדת בכוחות עצמך עם ההסברים בהמשך.
5. שים לב! בקטע מסוים של התהליך התרגיל שגוי. מצא את השגיאה, תקן אותה והפעל את המשפט מחדש.

תרגיל 1.15
קליטת נתונים במהלך מקרו פעיל

	A	B	C	D	E	F	G
1	\a {1_1}{getnumber Msg3,Read1} ~ {1_2}{quit}						
2	\b {1_1}{getlabel Msg4,Read1} ~ {1_2}{quit}						
3	1_1 {getnumber Msg1,NN1} ~ {getnumber Msg2,MM1} ~ {return}						
4	1_2 {put Data1,MM1-1,NN1-1,Read1} ~ {return}						
5	MM1			<--- מספר הסעיף התקציבי			
6	NN1			<--- מספר המחלקה			
7	Read1			<--- מה שרוצים לכתוב בלוח			
8	Msg1			--> מספר המחלקה (1..4) ואח"כ ENTER			
9	Msg2			--> מספר הסעיף (1 או 2) ואח"כ ENTER			
10	Msg3			--> נתונים (מספרים בלבד) ואח"כ ENTER			
11	Msg4			--> טקסט (בלבד) ואח"כ ENTER			
12							
13					לוח 1.15		
14				תקציבי מחלקות לפי סעיפים			
15							
16				סעיף תקציבי		מחלקה	
17				========================		==========	
18				1	2	#	שם
19				============	============	===	======
20							
21				25	500	1	בגדים
22	Data1 -->			15	350	2	וידיאו
23				12	275	3	ספרים
24				35	750	4	ספורט
25							

פקודת המקרו ~{getnumber Msg1,MM1}, בדומה לסימן השאלה, עוצרת את המחשב במהלך מקרו פעיל ומאפשרת למפעיל האנושי להחדיר מספר מבלי להפסיק את פעילות המחשב בשליטת המקורו. אלא שלא כבסימן השאלה מאפשרת פקודה זו אופציות שונות נוספות.

{getnumber Msg1,NN1}~

{getnumber	מלת הפעלה להחדרת נתון מספרי עם הפעלת פקודה זו תופיעה הודעה בשורה השניה של איזור הדו-שיח מעל הסרגל העליון (ראה עמוד 37). הודעה זו נקראת Prompt והיא מסבירה למפעיל המחשב באותו רגע מה המידע שהוא נדרש להזין בו את המחשב.
Msg1	הארגומנט הראשון (מופרד ממלת ההפעלה ע"י תו רווח) - שם תחום המכיל את תוכן ה Prompt. למשל <-- מספר המחלקה (1..4) ואח"כ ENTER. אפשר גם לכתוב את הטקסט עצמו בגרשיים כפולים בארגומנט הראשון במקום שם תחום - לא מומלץ. עם הופעת ההודעה באיזור הדו-שיח נעצרת פעילות המקרו והמחשב ממתין. עתה מתבקש המפעיל להקליד מספר ולהקיש ENTER. המספר יופיע באיזור הדו-שיח צמוד להודעה מימינה. לכן רצוי תמיד שהתו האחרון בצד ימין של ההודעה יהיה תו רווח.
NN1	הארגומנט השני היא הכתובת בגיליון בה יקלט המספר שהקלדנו. עם סיום הקלדת הנתון צריך להקיש ENTER.
}~	וכדי לעדכן את הגיליון מסתיימת הפקודה -->
{getlabel	אם מלת ההפעלה היא getlabel יקלט הנתון כטקסט ולא כמספר. נוהל זה דרוש כשרוצים לקלוט טקסט (למשל לשנות שם של יחידה) או לקלוט מספר כטקסט. נניח שרוצים לקלוט מספר בית שכתוב כך - 12/3. אם נשתמש ב getnumber נקבל ב NN1 את המספר 4 שהוא המנה מחילוק 12 ב 3. אך אם נשתמש ב getlabel נקבל את התפוקה הנכונה.

מה ההבדל בין שני המשפטים הראשיים בתרגיל הנוכחי.

במשפט b טמונה שגיאה מצא אותה ותקן.

נבחן עכשיו בצורה מסודרת את מבנה כל אחת מפיסקאות המקרו בתרגיל 1.15.

איור 1.12
בחינת מבנה פיסקת מקרו \a בתרגיל 1.15

	A	B	C	D	E	F	G
1	\a {1_1}{getnumber Msg3,Read1} ~ {1_2}{quit}						
2							
3	1_1 {getnumber Msg1,NN1} ~ {getnumber Msg2,MM1} ~ {return}						
4	1_2 {put Data1,MM1-1,NN1-1,Read1} ~ {return}						
5	MM1 <--- מספר הסעיף התקציבי						
6	NN1 <--- מספר המחלקה						
7	Read1 <--- מה שרוצים לכתוב בלוח						
8	Msg1 ENTER ואח"כ (1..4) מספר המחלקה -- >						
9	Msg2 ENTER ואח"כ (1 או 2) מספר הסעיף -- >						
10	Msg3 ENTER ואח"כ (מספרים בלבד) נתונים -- >						
11							
12				סעיף תקציבי			מחלקה
13				============	============	=====	======
14				1	2	#	שם
15				============	============	===	======
16							
17				25	500	1	בגדים
18	Data1 -- >			15	350	2	וידיאו
19				12	275	3	ספרים
20				35	750	4	ספורט

מטרת הפיסקה

קליטת מספרים בסעיפי התקציב של המחלקות השונות.

מבנה הפיסקה

הנתונים כתובים בתחום בשם Data1 שגודלו 4 עמודות X 4 שורות. יש ארבע מחלקות ושמותיהם בעמודה הראשונה מימין, שורה לכל מחלקה. בעמודה השניה מימין מספר סידורי. בשתי העמודות הראשונות משמאל נתונים תקציביים.

תחום בשם MM1 נועד לקלוט את מספר הסעיף התקציבי שיקבל את הנתון, התחום NN1 יקבל את מספר המחלקה והתחום Read1 יקבל את הנתון עצמו.

הפעילות מתבצעת ע"י שילוב {put ... }~ עם {getnumber ... }~ עם נוהל MsgX כאשר X = כל מספר.

סדר הפעולות

הפעלה	\a	מפעיל
קריאה למשפט שירות	{1_1}	
קליטת מספר המחלקה	{getnumber Msg1,NN1}~	שירות
Msg1 = <-- **מספר המחלקה** (1 .. 4) **ואח"כ** ENTER		
קליטת מספר הסעיף התקציבי	{getnumber Msg2,MM1}~	
Msg2 = <-- **מספר הסעיף** (1 **או** 2) **ואח"כ** ENTER		
חזרה למשפט המפעיל	{return}	
קליטת הנתון התקציבי (מספר)	{getnumber Msg3,Read1}~	מפעיל
Msg3 = <-- **נתונים** (**מספרים בלבד**) **ואח"כ** ENTER		
קריאה למשפט שירות	{1_2}	
העתקת הנתון אל Data1	{put Data1,MM1-1,NN1-1,Read1}~	שירות
חזרה למשפט המפעיל	{return}	
סיים	{quit}	מפעיל

ערוך ניתוח מבנה דומה למקרו השני. שים לב! הקטע השגוי נמצא כאן וממוקם ב נוהל Msg2. מצא אותו ונסה לתקנו בכוחות עצמך לפני שתמשיך.

מטרת משפט \b היא לאפשר שינוי טקסט בשם היחידה. לפיכך אין כאן מקום לאופציות בציון מספר העמודה בעת קליטת הטקסט החדש. תקן את המשפט בהתאם.

בוודאי מצאת את השגיאה ופיתחת מספר פתרונות. אנא זכור, העקרון הוא שכל מה שעובד זה טוב. אדגים כאן פתרון אחד שמטרתו לחדד את החשיבה המודולרית. הטכניקה תשמש אותנו פעמים רבות בהמשך.

בחן בתשומת לב את ההבדלים בין חיבור המקרו בתרגיל 1.15 לבין חיבור המקרו באיור 1.13. ערוך רשימת סדר פעולות לחיבור החדש. הסבר את תהליך החשיבה שהוביל לשינויים ואת הטכניקה. כיצד נעשה כאן שימוש במודולריות ובארגומנטים?

> אפשר לבצע מטלה מסוימת על ידי משפטי מקרו שונים. בעת הפיתוח רצוי לכתוב על פי העיקרון ש"מה שעובד זה טוב" ולא לנסות להגיע אל "השלמות" לפני שממשיכים.

איור 1.13
תיקון קטע שגוי בפיסקת מקרו \b בתרגיל 1.15.

	A	B	C	D	E	F	G
1	\a {1_1}{getnumber Msg3,Read1}~{1_2}{quit}						
2	\b {let Test1,1}~{1_1}{getlabel Msg4,Read1}~{1_2}{quit}						
3	1_1 {getnumber Msg1,NN1}~{1_3}{return}						
4	1_2 {put Data1,MM1-1,NN1-1,Read1}~{let Test1,0}~{return}						
5	1_3 @if(Test1=0,"{getnumber Msg2,MM1}~","{let MM1,3}~")&"{return}"						
6	MM1 <--- מספר הסעיף התקציבי						
7	NN1 <--- מספר המחלקה						
8	Read1 <--- מה שרוצים לכתוב בלוח						
9	Msg1 <-- מספר המחלקה (1..4) ואח"כ ENTER						
10	Msg2 <-- מספר הסעיף (1 או 2) ואח"כ ENTER						
11	Msg3 <-- נתונים (מספרים בלבד) ואח"כ ENTER						
12	Msg4 <-- טקסט (בלבד) ואח"כ ENTER						
13	Test1 1						

(חץ אל הנוסחה בשורה 5:) משוואה חיה

כשרוצים לקלוט נתון תקציבי (מספר) יש למפעיל אופציה לבחור באחת משתי עמודות וחובה עלינו לאפשר ברגע זה אינטראקציה בינו לבין המחשב. לעומת זאת כשרוצים לשנות שם של מחלקה אין אופציה כזאת מכיוון ששמות המחלקות מופיעים בעמודה אחת בלבד - MM1 = 3. ולכן מיותר כאן לשלב אינטראקציה אדם-מחשב וצריך לקלוט באופן אוטומטי MM1 = 3.

לשם כך יש ליצור נוהל בקרה אוטומטי שסדר הפעולות שלו יהיה כך:

1. בחירת אופציה - לקלוט נתון תקציבי או לקלוט שינוי שם מחלקה.
2. אם בחרנו קליטת נתון תקציבי כתוב 0 ב Test1 אחרת כתוב 1.
3. תן אפשרות לבחור את מספר המחלקה.
4. אם במצב (State) של קליטת נתון תקציבי (Test1 = 0) תאפשר לבחור מספר סעיף תקציבי, אחרת כתוב MM1 = 3.
5. תן אפשרות להקליד את הנתון שרוצים לקלוט.
6. קלוט את הנתון שהוקלד בכתובת שהוגדרה עבורו.
7. סיים.

הטכניקה לביצוע הבקרה האוטומטית הזאת ברורה למדי מהאיור שלמעלה, אך בכל זאת נסקור זאת בפרוט מלא.

מקימים תחום בשם Test1. יוצרים מצב כזה שאם מפעילים את משפט \b אזי Test1 יקבל את הסיפרה 1, אחרת יקבל את הסיפרה 0 (ראה באיור 1.13 את מיקום פקודות ה ~{let ... } שמבצעות את הפעולות הללו).

מקימים משפט שירות נוסף בשם 1_3. במשפט זה כותבים משוואה חיה המכילה התניה.

@if(Test1=0,"{getnumber Msg2,MM1}~","{let MM1,3}~")&"{return}"

הקם בגיליון שני תחומים אלה ואת MM1 והעתק את משוואת ההתניה לגיליון. כתוב ב Test 1 פעם את המספר 1 ופעם 0. ראה כיצד משתנה תפוקת משפט השירות כל פעם. שים לב שבכל מקרה מסתיימת תפוקת ההתניה בפקודה {return}. הגרשיים הכפולים דרושים כדי לכתוב טקסט בתוך משוואה חיה.

כדאי להתעמק עוד קצת בסוג חשיבה זה מכיוון שהוא חיוני להקמת אוטומציה יעילה.

הדבר דומה לטרמוסטט באקווריום. כאשר מצב המים הוא "קר" מפעיל הטרמוסטט את המחמם וכאשר מצב המים הוא "חם" מפסיק הטרמוסטט את פעולת המחמם. למעשה יש כאן שתי מערכות משולבות בזמן אמת. מערכת אחת יוצרת את החום הנדרש והמערכת המקבילה מספקת למפעילה את המידע הדרוש כדי להחליט מתי להפעיל את החימום ומתי להפסיקו.

כלומר, שתי המערכות המשולבות הללו מקיימות תהליך דינמי תוך בקרה עצמית אוטומטית השומרת על חום קבוע בגבולות מוגדרים מראש. שאם לא כן תיווצר הדרדרות המערכת והיא תתפרק ותחדל להתקיים (דיסאינטגרציה). למערכת כזאת אנו קוראים גם מערכת קיברנטית.

האינטרה מקרו נשוא הדיון הזה משלב שתי מערכות כאלה. האחת מבצעת את רצונותיו של המפעיל והשניה מקיימת מעקב מתמיד ואוטומטי אחרי המצב שנוצר בגיליון בעיקבות החלטות וביצועים תוך כדי התרחשותם. המידע הזה מוזרם אוטומטית אל המערכת המפעילה ומשנה פעילות מבלי לערב את המפעיל בהחלטה. כלומר הקמנו מנגנון לבקרה עצמית אוטומטית, בנינו מערכת קיברנטית.

בפיסקת המקרו הנוכחית יש שני משפטים מפעילים. בפרק ג נלמד להפעיל משפטים שונים בעזרת תפריט פקודות בעברית שנקים בעצמנו. לפעמים נצטרך להקים מערך קטן לידע את המחשב על פעולות שהחלטנו לבצע כדי להפעיל מנגנוני בקרה שונים.

1.4 למדנו בפרק זה

במקום n פעמים חץ למטה	{d n} = {down n}
במקום להקיש n פעמים חץ למעלה	{u n} = {up n}
במקום להקיש n פעמים חץ ימינה	{r n} = {right n}
במקום להקיש n פעמים חץ שמאלה	{l n} = {left n}
במקום להקיש n פעמים PgDn	{pgdn n}
במקום להקיש n פעמים PgUp	{pgup n}
במקום להקיש n פעמים Tab	{bigright n}
במקום להקיש n פעמים Shift Tab	{bigleft n}
במקום להקיש Home	{home}
במקום להקיש End	{end}
פונקציה במקום הארגומנט n	{d @n(Range)}
פונקציה במקום הארגומנט n	{d @rows(Range)}
שליפת נתונים ממטריצה	{d @index(Range,X,Y)}
מקרו בפונקציה - בתוך מקרו	@sum({d 2}.{end}{d})
מקרו להולכת סמן ע"י "קפיצת הדרך"	{goto}Range~
במקום מקש F9	{calc}
מחק את התוכן ב Range	{blank Range}~
קריאה למשפט שירות שנמצא ב Range	{Range}
סיום משפט שירות וחזרה למשפט הראשי	{return}
הקש ENTER	~
הפעלת משפט מקרו	Alt x x = any letter
הפעלת מקרו לאט ובשלבים - Step	Alt F2
הפסקת פעילות משפט מקרו לפני סיומו	Ctrl Break
שם של משפט מקרו	\a a = any letter
סיום משפט מקרו	{quit}
העתקת נתון מתא אל תא	{let Y,X}~
הודעות באיזור הדו-שיח	{indicate ... }
העתקת נתון אל תא במטריצת יעד	{put ... }~

העתקת נתון מתא במטריצת מוצא אל תא במטריצת יעד

{put Data1,MM1-1,NN1-1,@index(Data2,MM2-1,NN2-1}~

מקרו אינטראקטיבי	{?}

מקרו אינטראקטיבי	{getnumber ... }~
מקרו אינטראקטיבי	{getlabel ... }~

ולמדנו

לשלב ארגומנטים פשוטים ומורכבים במקרו
לכתוב חיבור מקרו במבנה מודולרי
לשלב כתיבת מקרו עם אירגון הגיליון לחיבור אינטרה מקרו אחד
לשלב מערכות הפעלה עם מערכות מידע להיזון חוזר לבקרה בזמן אמת - מערכת קיברנטית.

* להפסיק מקרו פעיל מפעילים את צמד המקשים - Ctrl Break

פרק ב: בסיסי נתונים ו Data Query

2.1 הנתונים והשטח

בחדר יש חפץ אחד בלבד - שולחן. על השולחן יש חפץ בודד - דף נייר חלק לגמרי. אומרים לי: "גש לחדר, בחדר יש שולחן, גש לשולחן וצייר על הנייר." התגובה הלוגית היחידה לשאלה זו היא: "לא הבנתי". חסרים נתונים - איפה החדר, איפה הנייר ומה לצייר. נאמר שזה חדר מספר שלוש במסגרת התייחסות שמגדירה את מיקומו יחסית למיקומי וכי אומרים לי: "גש לחדר מספר 3, בחדר יש שולחן, על השולחן יש נייר, צייר עליו בית". עכשיו, משקיבלתי נתונים מלאים ייצר המשפט שנאמר לי את התפוקה - בית מצוייר על דף הנייר היחידי שנמצא על השולחן היחידי בחדר מספר 3.

בדרך כלל הגדרת צרכים עונה על השאלה התמימה - מה רוצים לעשות. אלא שאי אפשר לענות עליה במלואה לפני שעונים על השאלה - איפה רוצים לעשות. למעשה במהלך התיכנון מחפשים בו-זמנית תשובות לשתי השאלות הללו ומהרגע הראשון עובדים עם סוגים שונים של נתונים.

תרגיל 2.01
נתונים ומשמעויות



סוג אחד של נתון היא הכתובת - בדוגמה זו מניחים שהוקמה תשתית שהיא מסגרת התייחסות המאפשרת להפוך את נתון הכתובת מרישום סתמי למידע בר משמעות. סוג שני עונה על השאלה - מה לעשות כשמגיעים לכתובת. לכך דרושה שפה להעברה חד ערכית של משמעויות מדובר אל שומע. סוג הנתון כאן היא המלה "צייר" והמידע הגלום בה היא המשמעות שמייחסים לה הדובר והשומע. סוג נוסף של נתון עונה על השאלה - מה לצייר, בדוגמה הנוכחית - "בית". יכל היה הדובר לבקש למשל "כתוב את המספר 325" או "כתוב את כמות המלאי הנוכחית של פריט מספר 73".

נאמר שכמות המלאי הנוכחית של פריט מספר 73 היא 325. כל הקורא את המספר 325 ללא הסברים נוספים לא יבין מזה ולא כלום. ואילו כשנכתוב לידו את המשפט - כמות המלאי הנוכחית של פריט 73 - מיד יקבל הנתון 325 משמעות מיוחדת. וראו כמה גילגולים עברנו כדי שנוכל להעביר משמעויות באמצעות נוהל קלט-פלט מסודר.

במשפטו של הדובר בתרגיל 2.01 טמונה הנחה. אם ההנחה נכונה יהיה ביצוע אחרת לא יהיה ביצוע. אם עדיין לא "קפצה" טעות זו לעיניך גלה אותה בדרך הבאה. רשום את סדר הפעולות הנדרש כדי שהשומע יוכל להבין בדיוק מה רוצים ממנו, מצא את החסר וכתוב את המשפט מחדש.

ההנחה הנסתרת במשפט הנוכחי היא שכאשר יגיע השומע לחדר 3 הוא יבין שיש שם שולחן, שעליו לגשת אליו, להתייצב ליד הנייר ולצייר על הנייר.

הנחה זו נכונה, אולי, כשמדובר בתיקשורת בין בני אדם. כי לפי הניסוח הנוכחי צריך להבין מהקשר הדברים והשטח שמציירים על נייר ולא על השולחן וכי לשם כך צריך לגשת לשם. אלא שהמחשב איננו אדם והוא יכול להפעיל חשיבה אינטואיטיבית לא סידרתית אך ורק אם נתכנת אותו לכך.

הנחה נסתרת נוספת היא שבחדר 3 יש שולחן אחד בלבד ועליו דף אחד בלבד וכי אפשר לצייר על הדף בכל מקום שרוצים.

לפיכך המשפט צריך להיות:

יש בשטח הנוכחי חדר ומספרו 3 ובו שולחן יחיד ועליו דף נייר יחיד. גש אל הדף הזה וצייר עליו בית.

שכן אם יש יותר משולחן אחד בחדר מספר 3 ועל כל אחד מהם יותר מדף נייר אחד חייבים לציין באופן חד ערכי את כתובת השולחן ובה את כתובת הדף בו רוצים לקלוט את הנתונים.

אם נבחן מקרוב את הדף נראה כי הוא חלק וריק לגמרי. יכולנו מראש להכין אותו כך שיהיו רשומים בו מספרי פריטים וליד כל פריט מקום ריק עבור הכמות הנוכחית במלאי. בראש עמודת הפריטים

נכתוב את המלה - פריט, ובראש העמודה הריקה שלידה נכתוב את התאור - כמות נוכחית במלאי. בכך נוספו נתונים על הנייר כבר בעת הקמת התשתית. לנתונים אלה יש משמעות בשתי רמות. בזמן קליטת הכמויות הנוכחיות במלאי יש כבר כתובת מדויקת לאיכסונן ובעת שליפת הנייר מהתיק יש מידע - כלומר, נתונים שיש להם משמעות.

```
          כמות
          נוכחית
פריט      במלאי
=====     =====

1
2
3
:
73
:
n
```

חשיבה בראשיתו של כל תיכנון היא תהליך דינמי ואינטואיטיבי המוליד תשובות לשאלות מה רוצים לעשות והיכן. לא ניתן להפריד בין השתיים ולעיתים קרובות כאשר המחשבה מדלגת במהירות בין "מה" ו"היכן" מתגלים האילוצים שמטילים שני אלה אחד על השני ומקבלים מושג טוב על המשך ההקמה.

2.2 מבנה לוחות

בפרק זה נעסוק בסוג חשוב של מאגרי נתונים אשר בו יש תפקיד מרכזי למבנה המחסן שמחזיק את הנתונים ולמבנה הדוח שמציג אותם - הלוח. נציג שני מבנים ונשווה ביניהם - בין מבנה לוח סגור לבין מבנה לוח פתוח.

2.2.1 לוח סגור

ללוח הסגור (ראה איור 2.01) מספר מאפינים:

1. הלוח סגור מכל עבריו על ידי קווים.
2. בין כל עמודת נתונים מפרידה עמודה המכילה קו אנכי.
3. עמודת סה"כ לכל פריט נמצאת בסוף הלוח משמאל.
4. השורה לסיכום העמודות היא האחרונה בתחתית הלוח.
5. כתוצאה מופיע הסה"כ הכללי בפינה השמאלית התחתונה של הלוח.

מבנה זה היה מקובל מאוד פעם ועדיין רואים אותו במספר מקומות ובמספר מקצועות. המבנה הסגור תואם להפליא את נוהל הסיכום הידני כפי שנלמד בבית הספר העממי ומשם, כנראה, נגרר אל

איור 2.01
מבנה לוח סגור

מכירות בדצמבר 1991 לפי פריטים ואיזורים

פריט		כמות		
מספר	תאור	איזור א	איזור ב	סה"כ
1	מחשב	560	9,077	9,637
2	רדיו	78	589	667
3	מיקסר	7,458	609	8,067
4	תוכנה	1,209	95	1,304
סה"כ		9,305	10,370	19,675

עיסוקים שונים בגיל מתקדם יותר. נוהל סיכום זה הוא:

```
   1
+  2
  ==
   3
  ==
```

והוא אמנם נוח לעבודה ידנית בכתיבת מספרים רבים ללוחות וסיכומם.

2.2.2 לוח פתוח

במשך השנים התרחב מגוון השימושים בנתונים ובלוחות ומבנה הלוח החל "להפתח" כדי להתאים לדרישות דינמיות יותר. אחד הדברים הראשונים היה שינוי מיקום עמודת ושורת הסה"כ. המשתמשים בלוח בדרך כלל מחפשים תחילה את הסה"כ הכללי ואח"כ את הסה"כ בראש העמודה או השורה. והנה מבנה הלוח הסגור מחזיק נתונים רבי משמעות אלה רחוק מהעין ולפיכך הועתק מיקומם והוצמד לכותרות (ראה איור 2.02).

פתיחת המבנה בוצעה לפני למעלה מ 03 שנה במספר מוסדות מחקר מכובדים בארה"ב. בתחילה נראה היה הדבר כתיקון קוסמטי בלבד אולם הוכח כיעיל ביותר עם כניסת המחשב האישי לשימוש נרחב. את הקווים האנכיים שגוזלים מקום רב השמיטו מהלוח

איור 2.02
לוח סגור - שיפורים במיקום נתוני סה"כ

מכירות בדצמבר 1991 לפי פריטים ואיזורים

פריט		כמות		
			איזור	
מספר	תאור	סה"כ	א	ב
	סה"כ	19,675	9,305	10,370
1	מחשב	9,637	560	9,077
2	רדיו	667	78	589
3	מיקסר	8,067	7,458	609
4	תוכנה	1,304	1,209	95

ונשארו הקווים האופקיים בלבד. מאחר שקווים אלה נועדו לסמן את תכולת העמודה ולא להדגיש את הכותרת נראה הלוח הפתוח כמו באיור 2.03.

איור 2.03
מבנה לוח פתוח

מכירות בדצמבר 1991 לפי פריטים ואיזורים

פריט		כמות		
			איזור	
מספר	תאור	סה"כ	א	ב
	סה"כ	19,675	9,305	10,370
1	מחשב	9,637	560	9,077
2	רדיו	667	78	589
3	מיקסר	8,067	7,458	609
4	תוכנה	1,304	1,209	95

בתחילה נועד מבנה הלוח להצגת נתונים בלבד. במהלך עשרות השנים האחרונות הלכו וגברו הדרישות וכשם שהשתפרו השיטות והנוהלים לאיכסון חפצים נולד צורך לפתח שיטות ונוהלים לאיכסון נתונים.

בגיליון האלקטרוני אפשר לבצע מספר סוגי פעולות על נתונים בלוח. את כל הפעולות הללו לא נוח ולא יעיל לבצע במבנה לוח סגור וקל ומהיר לבצען במבנה לוח פתוח. על פעולות אלה אנו מונים: מיון, הוספת וביטול שורות, הוספת וביטול עמודות עידכון, שליפה, עיבודים, פעולות מהסוג "אם ... אז ...", גראפים, הדפסות ועוד כהנה וכהנה.

2.3 Data Query

פעם היו ממלאים נתוני בלוחות, מדפיסים אותם ומפיצים. היום יש לנו בסיסי נתונים. בסיס נתונים מקימים כדי לענות על סוג מסוים של דרישות משותפות לצרכני מידע רבים. בספר זה נתייחס לבסיס נתונים תוך-גיליוני. בספרי הבאים, "שליטה ובקרה" "ומערכות מורכבות", אתייחס להקמת בסיס נתונים בדיסק לשליטה מתוך גיליון אלקטרוני.

בדרך כלל מתחילים ללמד את הנושא על ידי מתן הגדרות. הבה ננסה דרך לימוד שונה. נקים בסיס נתונים ותוך כדי מעשה נבחן מה עשינו.

איור 2.04 מציג לוח נתונים רגיל והעובר עליו בחטף יבין מיד במה מדובר. אילו היה הלוח ריק אזי שמו וכותרותיו, שהם בעצמם נתונים, נותנים למעיין שני סוגי מידע. הכותרות מגדירות את משמעות הנתונים ובאותה עת מציינות את מיקומם בבסיס הנתונים - את כתובתם בלוח. לפיכך מיקום הנתון בלוח (התא בו הוא שוכן) קובע את משמעותו ואת כתובתו.

בעגה המקצועית קוראים לעמודה שדה ולשורה קוראים - רשומה. הגדרות אלה תקפות רק כשהן מתייחסות למבנה כמו באיור 2.04 אשר בו משמעות כותרת השורה זהה בכל אחת מהעמודות המצטלבות בה ומשמעות כותרת העמודה זהה בכל אחת מהשורות המצטלבות בה.

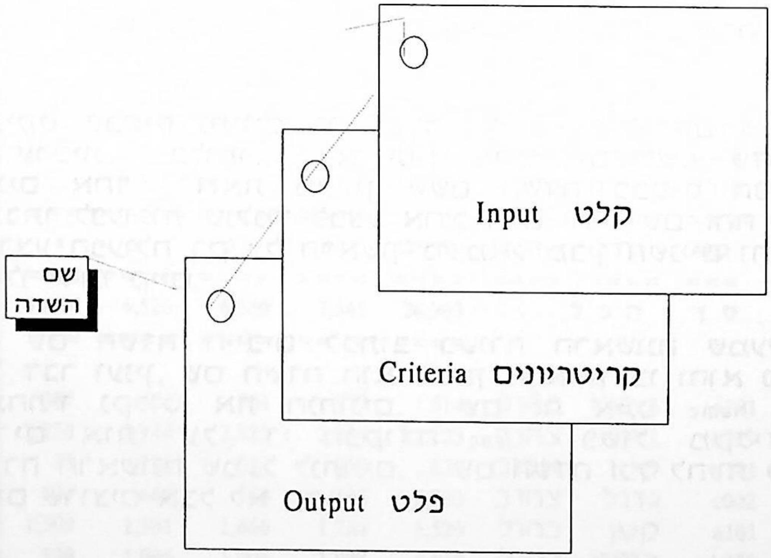
בגיליונות אלקטרונים מופעל בסיס נתונים תוך-גיליוני בנוהל המוגדר על ידי מספר קטן של פקודות תפריט באופציה Data Query /dq. הפקודה מאפשרת עידכון נתונים קיימים ושליפת נתונים ממקום אחד אל מקום אחר. לשם כך צריך להגדיר מראש את המקום אליו יקלטו הנתונים המקוריים, את המקום בו נכתוב מה רוצים לבצע ואת המקום אליו ישלפו הנתונים שביקשנו.

איור 2.04
שטח הקלט - מה שהמפעיל רואה

לוח 2.01 - מכירות לפי פריט ולפי איזורים

פריט									
מספר		תאור			איזור				
רץ	קט.	גודל	צבע	סה"כ	1	2	3	4
	סך	הכל		26,963	7,341	6,980	6,526	6,116
1	a101	בינוני	כחול	1,814	577	504	406	327
2	b079	בינוני	צהוב	11,123	2,834	2,825	2,744	2,720
3	a101	קטן	אדום	676	293	214	124	45
4	c002	גדול	צהוב	2,820	785	763	665	607
5	a101	קטן	כחול	6,520	1,764	1,666	1,581	1,509
6	b079	בינוני	כחול	4,010	1,088	1,008	1,006	908

בהקמת התחומים הללו יש לשמור על מספר כללים - כל איזור והכללים שלו. אבל לכולם יש גורם מבני אחד משותף שבלעדיו אי אפשר לקשר בין שלושת התחומים הללו - שם השדה.



2.3.1 תחום קלט

איור 2.05
תחום הקלט ומרכיביו

לוח 2.01 - מכירות לפי פריט ולפי איזורים

פריט					איזור			
מספר		תאור						
רץ	קט.	גודל	צבע	סה"כ	1	2	3	4
סך		הכל		26,963	7,341	6,980	6,526	6,116
x9	x8	x7	x6	x5	x4	x3	x2	x1
1	a101	בינוני	כחול	1,814	577	504	406	327
2	b079	בינוני	צהוב	11,123	2,834	2,825	2,744	2,720
3	a101	קטן	אדום	676	293	214	124	45
4	c002	גדול	צהוב	2,820	785	763	665	607
5	a101	קטן	כחול	6,520	1,764	1,666	1,581	1,509
6	b079	בינוני	כחול	4,010	1,088	1,008	1,006	908

נתונים

שם

רשומה ->

שדה x1   Dbase1   שדה x7

אוביקט בבסיס נתונים הוא שדה רק כשיש לו שם שמזהה אותו חד ערכית. כלומר, לא ינתן אותו שם לשני שדות בבסיס נתונים אחד. וזאת מכיוון ששם השדה בבסיס הנתונים הוא כתובתו לפעולות שנלמד לבצע אח"כ ואם יהיה שם זהה לשני שדות תתבצע הפעולה רק על הראשון משמאל מבין השניים והשדה הימני כאילו איננו קיים.

את שם השדה חייבים לכתוב בשורה הראשונה שמעל לנתונים. לכל דבר ועניין, שם השדה הוא הנתון הראשון בו והוא נשאר קבוע ומתחתיו נקלוט את הנתונים. שם זה איננו Range Name ולא יוצרים אותו על ידי הפקודה /rnc. פשוט מקלידים אותו בשורה הראשונה שמעל לנתונים. שם השדה יכל להיות כל רצף של תווים שרוצים אבל לא מספר.

נוהל יעיל וטוב למתן שמות שדות הוא השימוש באות אחת ומספר צמוד אליה

x1 x2 x3 x4 x5 x6 x7 x8 x9

בצורה זו לא יפגעו מבנה הלוח וצורת הצגת הנתונים וקל לשמור על חד ערכיות. כדי שהשמות הללו לא יראו בגיליון אחרי כתיבתם וגם לא יודפסו בדו"ח נחביא אותם על ידי הפקודה Range Format Hidden. השמות לא יראו אבל המחשב לא יתעלם מהם בעת הפעלת בסיס הנתונים. כדי לראות שוב את השמות מבצעים Range Format Reset.

ועתה, משגמרנו להכין את כל אלה נגדיר את כל השטח המוצל שרואים באיור 2.05 בשם Dbase1. לשם כך נשתמש בפקודה Range Name Create ונצבע את השטח הכולל את כל שמות השדות ואת כל הנתונים שמתחתם. שם זה - Dbase1 - יהיה אח"כ הארגומנט שיגדיר למחשב את איזור הקלט משתופעל הפקודה Data Query.

העתק את הלוח מאיור 2.05 לגיליון במחשב שלך. מקם אותו בגיליון כמודגם בתרגיל 2.02 והגדר את כל הדורש הגדרה כתשתית לפני הפעלת הפקודה /dq. אל תשמיט שום פרט, גם מה שנדמה כ"סתם קוסמטיקה" או כשורת רווח מיותרת.

תרגיל 2.02
מיקום נתונים לקראת המשך הפיתוח

	A	B	C	D	E	F	G	H	I	J
1	לוח 2.01 - מכירות לפי פריט ולפי איזורים									
2										
3										
4										
5							פ ר י ט			
6						=====	=====	=====	=====	
7		א י ז ו ר				ת א ו ר		מ ס פ ר		
8	=====	=====	=====	=====		=====	=====	=====	=====	
9	4	3	2	1	סה"כ	צבע	גודל	קט.	רץ	
10	===	====	====	====	====	====	====	===	====	
11	6,116	6,526	6,980	7,341	26,963	ה כ ל		ס ך		
12	===	====	====	====	====					
13										
14	327	406	504	577	1,814	כחול	בינוני	a101	1	
15	2,720	2,744	2,825	2,834	11,123	צהוב	בינוני	b079	2	
16	45	124	214	293	676	אדום	קטן	a101	3	
17	607	665	763	785	2,820	צהוב	גדול	c002	4	
18	1,509	1,581	1,666	1,764	6,520	כחול	קטן	a101	5	
19	908	1,006	1,008	1,088	4,010	כחול	בינוני	b079	6	

2.3.2 כיצד המחשב חושב

קטע זה נועד להסביר כיצד פועלת פקודת Data Query מהרגע שהופעלה. אין כאן תרגילים מכיוון שמה שיוסבר לא נועד לביצוע על ידי המשתמש אלא להבנה בסיסית.

מהרגע שמפעילים את הפקודה Data Query מתחיל המחשב להיות מאוד מאוד עסוק, טרוד ומתרוצץ. כדי להבין את המתרחש נבחן את סדר הפעולות שלו.

לא משנה מה רוצים שתהיה התוצאה, הדבר הראשון שהמחשב עושה עם הפעלת הפקודה זה חיפוש רשומות בתוך ה Input Range. בתרגיל הנוכחי הגדרנו תחום בשם Dbase1 ומכאן ואילך יהיה הוא איזור הקלט שלנו. ובכן המחשב מחפש רשומות ב Dbase1 וכאן הוא נתקל בבעיה - איזה רשומה לחפש. זאת צריך להגיד לו לפני שמבקשים ביצוע - ותהיה מטרת החיפוש אשר תהיה.

יש מספר שיטות חיפוש. עד כה עסקנו בהרחבה בנוהל חיפוש כתובת ומעט בנוהל lookup. כאן נעסוק בנוהל שונה.

משל לאדם שבא לעיר זרה ובידו תיאור של בית שגר בו ידיד שכתובתו איננה ידועה לו. משוטט בעיר ופוגש שוטר נחמד ומבקש את עזרתו. אילו כתובת בידו יוכל להגיע אל בית ידידו ולא למקום אחר. אולם עתה עליו לנסות ולתאר את הבית כמיטב יכולתו. אם יחפש בית אדום יעלה רבים בחכתו. יוסיף ויאמר - אדום, בן קומה אחת, בשפתו השמאלית של הנהר, ליד גשר הפרחים - יקבל מעט כתובות ואולי כתובת אחת בלבד אם עלה מזלו. ואפשר גם שלא ימצא אם תעה ולא זו העיר.

אפשר לאמר שהמחפש מתאר פרופיל של בית חברו. פרופיל זה מורכב מצרוף תכונות שונות ומאפיינים שונים:

פ ר ו פ י ל   ק ר י ט ר י ו נ י ם

	מאפיין (שדה)			תכונה (נתון)
	==========			======
מה ?	מבנה	איזה	?	בית
מה ?	צבע	איזה	?	אדום
מה ?	קומות	כמה	?	1
מה ?	שפת נהר	איזה	?	שמאל
מה ?	גשר	איזה	?	הפרחים

כמובן שלבית הידיד יש עוד מאפיינים (שדות) למשל: רחוב ומספר בית, יישוב ומיקוד, טלפון, מספר דירות, מספר דיירים, גיל ממוצע של בעלי עיניים כחולות וכל מה שתרצו.

במילים אחרות, יש למחפש רשימה של ידידים. ברשימה זו לכל ידיד שורה ובה מקומות (שדות) לרשום נתונים. חלק מהמאפיינים הללו, כמו צבע הבית, רלבנטיים למציאת הבית והיתר אין להם כל שייכות למיקומו. את פרופיל הקריטריונים מספקים למחשב לפני הפעלתו. כיצד ל"צייר" פרופיל קריטריונים נלמד בהמשך ובינתיים נניח שהמחשב קיבל את המידע. דהיינו, המחשב יודע היכן איזור הקלט ומה הוא פרופיל הקריטריונים.

מה שמבצע המחשב אפשר אולי לתאר כדלקמן.

1. המחשב אומר: יהי משפט מספר 1 ובו כתוב -

שדה מבנה=בית וגם שדה צבע=אדום וגם שדה קומות=1
וגם שדה שפת נהר=שמאל וגם שדה גשר=הפרחים.

2. המחשב ניגש לרשומה הראשונה בבסיס הנתונים (לא משנה סדר הרשומות) ואומר כך

משפט מספר 1 נכון לגבי רשומה זו.

3. אמירה זו של המחשב, או שהיא נכונה או שהיא לא נכונה. את התשובה רושם לו המחשב במקום כל שהוא כך

אם נכון אז 1
אחרת 0

4. המחשב עובר לרשומה הבאה ולכל רשומה שאחריה עד לרשומה האחרונה ברשימה וחוזר כל פעם על השאלה ועל סימון התשובה אצלו.

בכך סיים המחשב "לחשוב". הוא כבר יודע היכן - אצלו - יש רשומות שמתאימות לפרופיל הקריטריונים אבל עדיין איננו יודע מה לעשות עם הרשומות שמצא.

2.3.3 פרופיל קריטריונים ותחום קריטריונים

דרוש מקום בכתובת מוגדרת מראש כדי לרשום את פרופיל הקריטריונים. נקרא למקום הזה Crt1 והמחשב יקבלו כארגומנט בעת הפעלת הפקודה. זה המקום בו יחפש המחשב את פרופיל הקריטריונים. נתייחס אליו במונח תחום הקריטריונים - Criteria Range.

איור 2.06
תחום הקלט Input Range

	A	B	C	D	E	F	G	H	I	J
1	לוח 2.01 - מכירות לפי פריט ולפי איזורים									
2				ק ל ט						
3										
4										
5							פריט			
6				Dbase1		=====	=====	=====	=====	
7		איזור				תאור		מספר		
8	====	====	====	====		====	====	====	====	
9	4	3	2	1	סה"כ	צבע	גודל	קט.	רץ	
10	===	====	====	====	====	====	====	===	====	
11	6,116	6,526	6,980	7,341	26,963		סך הכל			
12	===	====	====	====	====					
13	x1	x2	x3	x4	x5	x6	x7	x8	x9	
14	327	406	504	577	1,814	כחול	בינוני	a101	1	
15	2,720	2,744	2,825	2,834	11,123	צהוב	בינוני	b079	2	
16	45	124	214	293	676	אדום	קטן	a101	3	
17	607	665	763	785	2,820	צהוב	גדול	c002	4	
18	1,509	1,581	1,666	1,764	6,520	כחול	קטן	a101	5	
19	908	1,006	1,008	1,088	4,010	כחול	בינוני	b079	6	

נאמר שמנהל האגף רוצה לקבל מיד את המכירות לפי פריטים ולפי איזורים - אך רק לפריטים שגודלם בינוני וצבעם כחול. פרופיל הקריטריונים הנובע מדרישה זו הוא

קישור לוגי

שדה x7 = בינוני וגם שדה x6 = כחול

נעבור לעמודה I וממנה וימינה נקים איזור לפרופילים לפי הדוגמה באיור 2.07. את שמות השדות חייבים להעתיק ע"י פקודת Copy כדי ששמות השדות בתחום הקריטריונים יהיו זהים לשמות באיזור הקלט. עכשיו נגדיר /rncCrt1 את השטח שכולל את שורת שמות השדות ושורה אחת נוספת כמתואר באיור 2.07. בתא שמתחת ל x7 נכתוב את המלה - בינוני, ומתחת ל x6 נכתוב - כחול. איות המילים והרווחים חייבים להיות זהים עם תואמיהם באיזור הקלט. כשנפעיל את הפקודה יבין המחשב כי

1. פרופיל הקריטריונים נמצא ב Crt1
2. המאפיינים הרלבנטיים הם שדה x7 ושדה x6
3. התכונות המבוקשות הן בינוני וכחול
4. הקישור הלוגי בין התכונות המבוקשות הוא - גם

איור 2.07
תחום הקריטריונים: הקישור הלוגי - גם

	I	J	K	L	M	N	O	P	Q	R
1	לוח 2.01 - מכירות לפי פריט ולפי איזורים									
2	פרופיל קריטריונים									
3										
4										
5							פריט			
6	Criteria Range =		Crt1			====	====	====	====	
7		איזור					תאור	מספר		
8	====	====	====	====	====	=====	=====	====	====	
9	4	3	2	1	סה״כ	צבע	גודל	קט.	רץ	
10	===	====	====	====	====	====	====	===	====	
11										
12										
13	x1	x2	x3	x4	x5	x6	x7	x8	x9	
14						כחול	בינוני			
15										
16										
17										

כאשר רושמים את התכונות של שדות שונים באותה שורה בתחום הקריטריונים (שורה 14 בדוגמה הנוכחית) מבין מכך המחשב שהקישור הלוגי ביניהן הוא המלה - גם. לפיכך יבין המחשב את פרופיל הקריטריונים בדיוק כפי שרצינו ובמהלך החיפוש יסמן את כל הרשומות שלגביהן פרופיל זה נכון (רשומה 1 ורשומה 6 בדוגמה).

תרגיל 2.03
תחום הקריטריונים: הקישור הלוגי - או

1. בשדות שונים

5							פריט		
6					====	====	====	====	
7		איזור					תאור	מספר	
8	====	====	====	====	====	=====	=====	====	====
9	4	3	2	1	סה״כ	צבע	גודל	קט.	רץ
10	===	====	====	====	====	====	====	===	====
11									
12									
13	x1	x2	x3	x4	x5	x6	x7	x8	x9
14							בינוני		
15						כחול			

כשרוצים להפעיל את הקישור הלוגי - או - כותבים את התכונה של המאפיין (שדה) האחד בשורה אחת ואת תכונת המאפיין (שדה) האחר בשורה שונה לפי הדוגמה בתרגיל 2.03.

שים לב שתחום הקריטריונים מכיל עכשיו שלוש שורות.

כתוב את פרופיל הקריטריונים של תרגיל 1 - 2.03

איזה רשומות יסמן המחשב בעת החיפוש?

שים לב...

שורה ריקה בתחום הקריטריונים גורמת

לסימון כל הרשומות באיזור הקלט

לפיכך, פרט לביצוע מטלה אחת שנלמד עליה בהמשך,

נקפיד תמיד

לא לכלול שורות ריקות

בתחום הקריטריונים.

תרגיל 2.03
תחום הקריטריונים: הקישור הלוגי - או

2. בשדה אחד

5	פריט								
6	====	====	====	====	====				
7	מספר		תאור			איזור			
8	====	====	====	====		====	====	====	====
9	רץ	קט.	גודל	צבע	סה"כ	1	2	3	4
10	====	====	====	====	====	====	====	====	===
11									
12									
13	x9	x8	x7	x6	x5	x4	x3	x2	x1
14			בינוני						
15			גדול						

כשכותבים שתי תכונות בשדה אחד יתרחש הקישור הלוגי - או - ביניהן והמחשב יסמן את כל הרשומות שגודלן בינוני או גדול (1,2,4,6).

כתוב את פרופיל הקריטריונים של תרגיל 2 - 2.03

תרגיל 2.03
תחום הקריטריונים: הקישור הלוגי - ו/או
3. משולב

מנהל האגף ביקש ריכוז נתוני הפריטים הגדולים ואת אלה שבגודל בינוני וגם כחולים. הוא ביטא זאת במילים שלו כך

תן לי את ריכוז הפריטים הגדולים יחד עם הכחולים בינוניים.

כתוב את פרופיל הקריטריונים.
הקם ומלא את תחום הקריטריונים.
איזה רשומות יסמן המחשב.

פרופיל הקריטריונים: (שדה x7 = בינוני וגם שדה x6 = כחול) או שדה x7 = גדול

תחום הקריטריונים -->

x7	x6
בינוני	כחול
גדול	

יסומנו רשומות 6 4 1

מנהל האגף ביקש

תן לי את הגדולים ואת הכחולים.

כתוב את פרופיל הקריטריונים.
הקם ומלא את תחום הקריטריונים.
איזה רשומות יסמן המחשב.

פרופיל הקריטריונים: שדה x6 = כחול או שדה x7 = גדול

תחום הקריטריונים -->

x7	x6
	כחול
גדול	

יסומנו רשומות 6 5 4 1

את כותרות הלוח כותבים בשטח הקריטריונים
בגיליון כדי להקל על ההתמצאות

עד כה עדיין לא נאמר למחשב מה הוא צריך לעשות עם הרשומות שמצא. לפני שתמשיך לקרוא הכן בעצמך רשימת מטלות שאפשר לדעתך להטיל על המחשב לבצע עם הרשומות התואמות את פרופיל הקריטריונים.

משהגדרנו למחשב את המטלה לביצוע עם רשומות תואמות פרופיל הקריטריונים הושלמה הגדרת השאילתא והמחשב עובר אוטומטית לביצוע. את התוצאות נקבל בתחום הפלט.

אפשר לאמר שהשאילתא היא ליבו של בסיס הנתונים שכן בסיס נתונים הוא מבנה למטרת הפעלת נוהל שאילתות - ולא סתם מחסן סגור - אין יוצא ואין בא. והשאילתא היא הקשר בין הצרכן למחשב. בדוגמה שלנו הצרכן הסופי הוא מנהל האגף ולא המפעיל או המתכנת ובהכנת השאילתא מוגשת לנו דוגמה חשובה לנוהל תיקשורת אדם מחשב.

1. תן לי כחולים או גדולים
2. פרופיל קריטריונים
3. מטלה לבצע
4. פלט

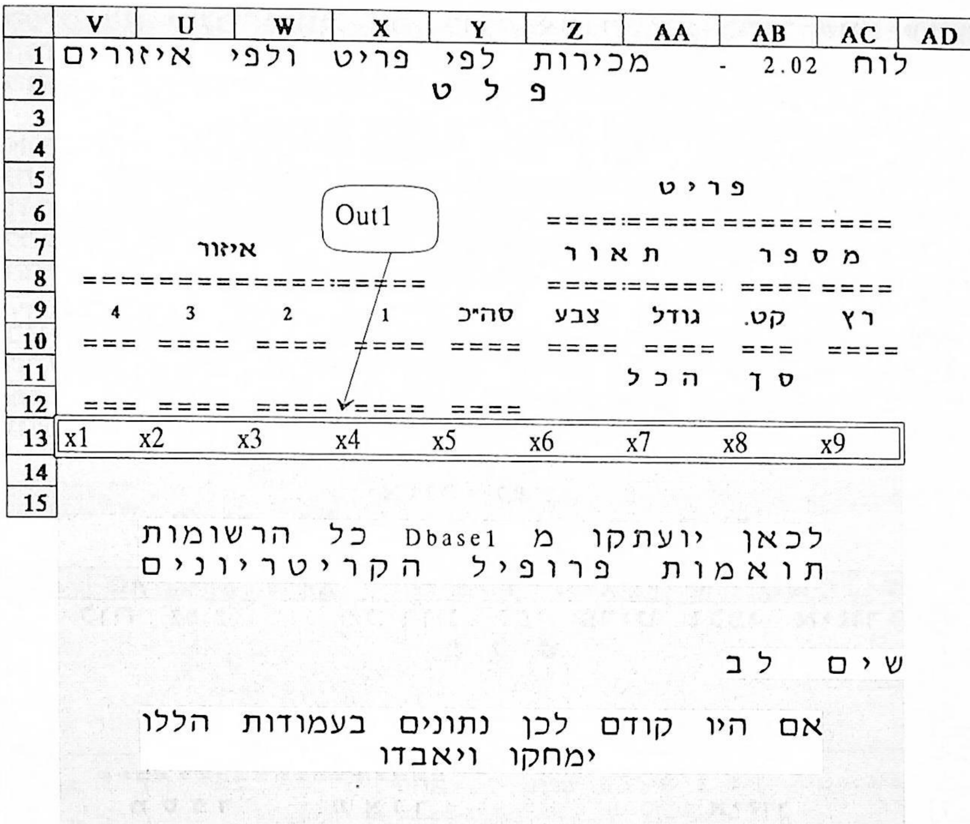
סדר פעולות זה בנוהל תיקשורת אדם מחשב ניתן לביצוע בתנאי שהוקמה תשתית מתאימה. כאן חסר עדיין מרכיב חיוני, אולי החיוני ביותר. כשמבקש המנהל - תן לי כחולים או אדומים - עדיין אין יודעים מה רוצה הוא לראות בהם, חסרה הגדרת הצרכים של משתמש הקצה. את הגדרת הצרכים של משתמש הקצה צריך להכין בצורה כזאת שכאשר נגיש לו את החומר תהיה תגובתו "יופי, זה בדיוק מה שרציתי" ולא "לא הבנתם אותי".

2.3.4 תחום פלט

נאמר שהמטלה המתבקשת היא לשלוף מתוך Dbase1 את כל הרשומות תואמות פרופיל הקריטריונים. ראינו גם שהחוט המקשר בין תחום הקלט (Input Range) לבין תחום הקריטריונים (Criteria Range) הם שמות השדות. ושמות השדות הם גם המרכיב החיוני של תחום הפלט (Output Range). ממבנה התשתית עד כה רואים שבכל אחד משני התחומים האחרים, הקלט והקריטריונים, שורת שמות השדות תמיד ראשונה, וכך גם בתחום הפלט. נאמר שבחרנו לקרוא לתחום הפלט בשם Out1. נצטרך לבצע זאת על ידי הפקודה /rncOut1 וצביעת השטח המוגדר. כאן יש שתי אופציות ונתייחס אל כל אחת בנפרד.

נתבונן באיור 2.08 ונתעלם לרגע מהכל פרט ל Out1. תחום זה נמצא בעמודות שמימין לעמודות האחרונות של תחום הקריטריונים. הוא מכיל רק את שמות השדות ולפיכך כולל שורה אחת בלבד. כאשר

איור 2.08
תחום פלט (Output Range) פתוח



נפעיל את המחשב לשליפת רשומות תואמות פרופיל קריטריונים יתבצע סדר הפעולות הבא.

1. המחשב יגש אל התחום Dbase1
2. יבחן את כל הרשומות - ללא יוצא מהכלל - ויסמן את הרשומות תואמות פרופיל הקריטריונים.
3. אם אין אפילו רשומה תואמת אחת נקבל הודעת ERROR, אחרת
4. ימחק המחשב כל מה שנמצא בעמודות שהוגדרו בתחום הפלט החל בשורה הראשונה שמתחת לשורת שמות השדות ועד לשורה התחתונה בגיליון - 8192
5. המחשב יעתיק את כל הרשומות המסומנות ויכתוב אותן בשטח שנמחק זה עתה החל בשורה הראשונה מתחת לשמות השדות. תא בתחום הקלט שיש בו נוסחה יועתק הערך ולא הנוסחה.

כאן מגיעים לאופציה חשובה ביותר לשליפת הנתונים. מכל אחת מהרשומות שסומנו ישלפו הנתונים מהשדות ששמותיהם כתובים ב Out1 ובעמודה שמתחתיו. לפיכך אם לא נכתוב את שם השדה בתחום הפלט לא ישלפו נתוניו, אם נכתוב את השמות בסדר שונה יופיעו הנתונים לפי הסדר החדש מבלי שהדבר ישפיע כלל על מהירות הביצוע ואם נשאיר עמודה ללא שם שדה נקבל עמודה ריקה.

אופציה זו מעניקה לנו מכשיר טוב לעיצוב מבנה הפלט. את נתוני הקלט שומרים במבנה צפוף, אחיד וללא משוואות. לכל דוח מעצבים מראש מבנה לוח עבור הפלט שרוצים (ראה איור 2.08), להשאיר בו עמודות ריקות לחישובים (למשל סה"כ) ולבצע את החישובים אחרי השליפה בנוהלים שלמדנו בפרק א בספר זה.

באיור הנוכחי רואים גם מבנה לוח מוכן להדפסה. אחרי הקמת התשתית נעלים את שורת שמות השדות בתחום הפלט ע"י הפקודה /rfh. כתוצאה מכך לא יופיעו השמות הללו בתדפיס. עוד על נושא אירגון החומר לשליטה ולבקרה בפרק האחרון.

איור 2.09

תחום פלט (Output Range) סגור

	V	U	W	X	Y	Z	AA	AB	AC	AD
1	לוח 2.02 - מכירות לפי פריט ולפי איזורים									
2					פ ל ט					
3										
4										
5								פ ר י ט		
6						====	====	====	====	
7		איזור				תאור		מספר		
8	====	====	====	====		====	====	====	====	
9	4	3	2	1	סה"כ	צבע	גודל	קט.	רץ	
10	===	====	====	====	====	====	====	===	====	
11							הכל	סך		
12	===	====	====	====	====					
13	x1	x2	x3	x4	x5	x6	x7	x8	x9	
14										
15					Out1					
16										

אם יש נתונים כל שהם בין איזור הפלט לבין תחתית הגיליון ורוצים למנוע את מחיקתם בעת הפעלת השאילתא מגדירים את תחום הפלט לפי הדוגמה באיור 2.09. כאשר מגדירים /rncOut1 על יותר משורה אחת מגבילים מראש את מספר הרשומות המסומנות שישלפו למספר השורות ב Out1 פחות 1. כאן נוצרת בעיה

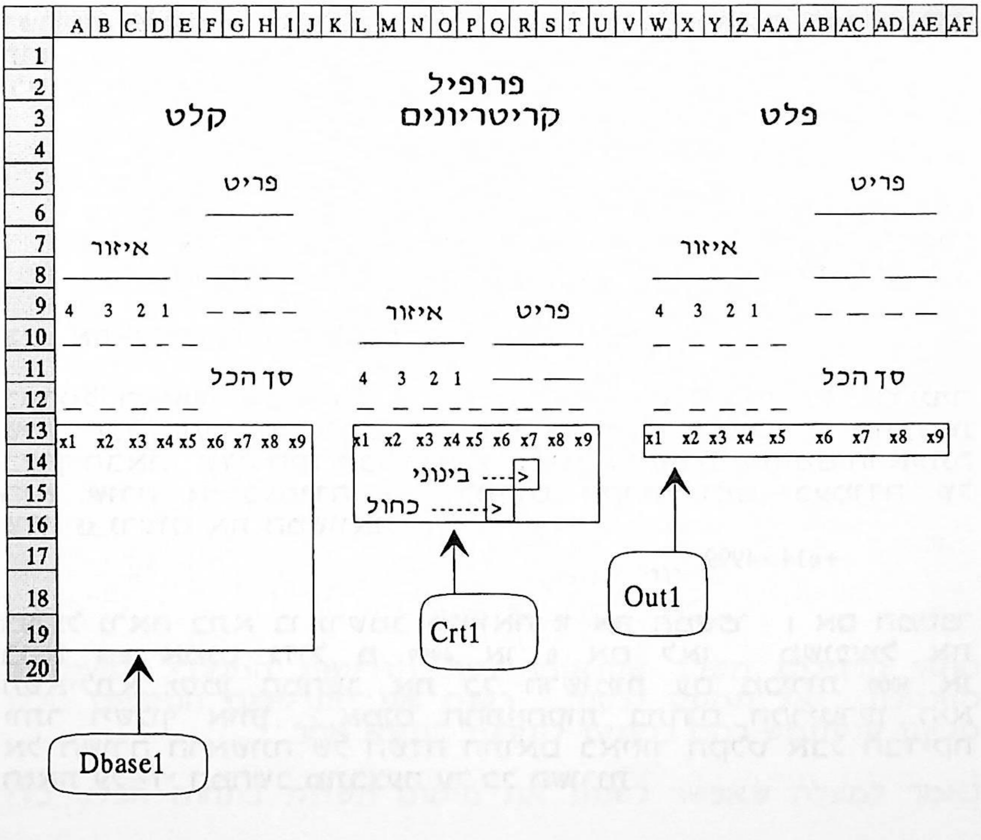
חמורה ביותר - אם סומנו באיזור הקלט מספר רשומות תואמות קריטריונים יותר ממספר השורות שהוכן עבורן באיזור הפלט ישלפו רק הראשונות והדיווח יהיה שגוי. גם הגדרת Out1 מראש לפי גודל Dbase1 איננה עונה על הבעיה מכיוון שתחום הקלט עשוי להיות דינמי וגודלו ישתנה מפעם לפעם ע"י הוספה או ביטול של רשומות. לפיכך

מומלץ להשתמש תמיד בנוהל תחום פלט פתוח.

בתרגיל 2.04 מוצג דגם של גיליון מאורגן במבנה ל Data Query. איור זה מרכז במבט אחד את כל המרכיבים שנדונו למעלה - קלט, פרופיל קריטריונים ופלט.

בהמשך לתרגיל הקודם הקם את כל המבנה הזה בגיליון והגדר את התחומים כמתואר בתרגיל 2.04. העתק את כל הנתונים לגיליון.

תרגיל 2.04
מבנה גיליון ל Data Query



בתחום הקריטריונים הגדר את הפרופיל

גודל = בינוני או צבע = כחול

לפי הדוגמה שבתרגיל 2.04.

עכשיו הקם והפעל את המקרו שבאיור 2.10 (שים לב למיקומו בגיליון).

איור 2.10
מקרו לשליפה ב Data Query

	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1		\z	/dqiDbase1~cCrt1~oOut1~eq{quit}							
2										

אם לא הצלחת בדוק את הגדרות התחומים ואת הזהות בשמות השדות בין התחומים השונים ונסה עד שתצליח.

תרגיל מספר 2.05 מורכב מסידרה של תחומי קריטריונים לשליפת רשומות שונות. העתק את הראשון לגיליון, התאם את הגדרת תחום הקריטריונים (ע"י /rncCrt1) שיכלול את כל השורות הדרושות ורק אותן והפעל את המקרו \z. אחר כך

1. בחן את הרשומות שנשלפו ואת אלה שלא נשלפו
2. כתוב את משפט פרופיל הקריטריונים
דמיין שהדוח מיועד למי שאיננו מומחה בניסוח קריטריונים. נסח מחדש את פרופיל הקריטריוני בלשון "אנושית" לא פורמלית למשל -
3. תן לי את הגודל הקטן של המספרים הקטלוגיים שמתחילים ב a.

בצע את הפעולות הללו לכל אחד מהפרופילים בתרגיל זה.

בתרגיל התשיעי שבסידרה זו אנו מבקשים לשלוף את כל הפריטים שסך הכל המכירות שלהם היה 5000 או יותר. פעולה זו מתבצעת בדרך הבאה. שדה הסך הכל הוא x5. בשדה זה שורת הנתונים הראשונה היא שורה 14 בעמודה e. בתחום הקריטריונים, בעמודה של שדה x5 נרשום את המשוואה

+e14>4999

בפועל נראה בתא בו נרשמב משוואה זו את המספר 1 אם המספר בתא e14 אמנם גדול מ 4999 או 0 אם לאו. משנפעיל את השאילתא יסמן המחשב את כל הרשומות עם מכירות 5000 או יותר וישלוף אותן. אמנם ההתייחסות בתחום הקריטריון היא אל השורה הראשונה של השדה התואם באיזור הקלט אבל הבדיקה הזאת על ידי המחשב מתבצעת על כל השורות.

תרגיל 2.05
כתיבת פרופיל קריטריונים ל Data Query

.1

x1	x2	x3	x4	x5	x6	x7	x8	x9
						בינוני		
						גדול		

.2

x1	x2	x3	x4	x5	x6	x7	x8	x9
								4

.3

x1	x2	x3	x4	x5	x6	x7	x8	x9
								2
								3
								5

.4

x1	x2	x3	x4	x5	x6	x7	x8	x9
				(שורה ריקה)				

.5

x1	x2	x3	x4	x5	x6	x7	x8	x9
								a*

.6

x1	x2	x3	x4	x5	x6	x7	x8	x9
								?0??

.7

x1	x2	x3	x4	x5	x6	x7	x8	x9
						קטן		a*

.8

x1	x2	x3	x4	x5	x6	x7	x8	x9
						קטן		a*
					צהוב			

.9

x1	x2	x3	x4	x5	x6	x7	x8	x9
				+e14>4999				

.10

x1	x4	x7
	+d14>1499	בינוני
+a14>=50		קטן

הוסף תרגילים משלך שיתאימו לסוג הדרישות המקובל בפעילותך השוטפת. אל תהסס לנסות. רשום את התרגילים הנוספים הללו מכיוון שנשתמש בהם בהמשך לתירגול בנושא אחר.

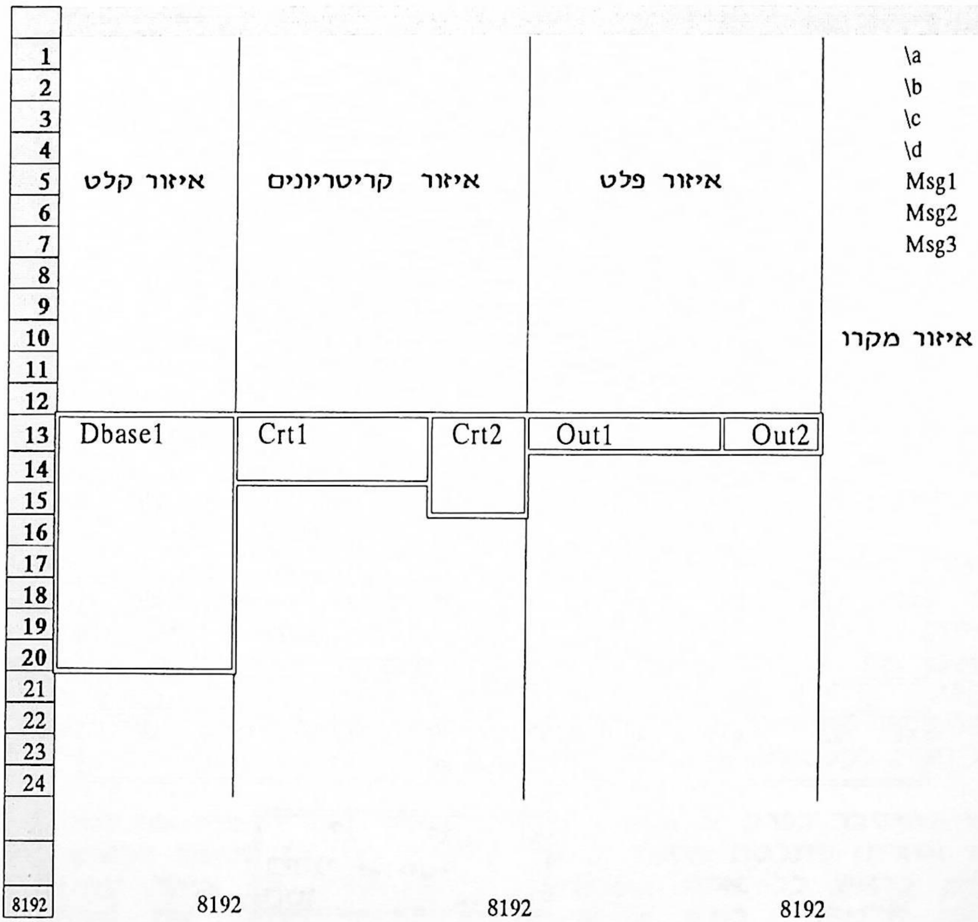
נאמר למעלה שאפשר לשנות את מיקום השדות בתחום הפלט כדי

לקבל דוחות שונים. נבצע עתה סידרה של תרגילים שימחישו כיצד אפשר, בגיליון אחד, להכין לפני אוטומציה מספר דוחות השואבים מידע מתחום קלט מרכזי אחד.

באיור 2.11 אנו רואים מבט כללי על גיליון המכיל מספר תחומי קריטריונים ומספר תחומי פלט המתייחסים לאותם נתוני קלט.

איור 2.11

תשתית מורחבת לאוטומציה ב Data Query



איור זה מדגיש את חלוקת הגיליון לאיזורי פעילות ואת חלוקת האיזור לתחומי משנה. מומלץ לשמור, למען הסדר הטוב, שאיזורים ותחומים בגיליון לא יחצו האחד את השני. במילים אחרות - מוטב לוח ליד לוח ולא לוח מתחת ללוח.

במהלך הפעילות השוטפת אנו עושים שימוש חוזר בפרופיל קריטריונים אחד או יותר. במקרה כזה רצוי לחלק את איזור הקריטריונים למספר תחומים (Crt1, Crt2, ... CrtN) כמספר הפרופילים שיש לנו. נגדיר כל אחד מהפרופילים הללו פעם אחת בעת הקמת התשתית ואח"כ נוכל ללא קושי להפעילם ע"י מקרו. אם נשמור על הכלל של מקביליות האיזורים והתחומים נוכל תמיד להוסיף תחומים ע"י הכנסת עמודות היכן שנרצה מבלי לשבש איזה שהוא דבר.

אותו עקרון ישמש אותנו באיזור הפלט וכאן חשיבותו רבה ביותר. ראינו למעלה שאפשר לשלב באיזור הפלט תבנית לוח מוכן להדפסה עם תחום לשליפת נתונים ועיבודם. כתוצאה מכך יניב משפט מקרו קצר אחד שליפת רשומות תואמות פרופיל ישר אל מבנה הלוח ומיד אחר כך את הדפסתו בפורמט סופי. נוכל להפעיל אוטומציה גם להנפקת דוחות שונים מנתוני אותו הקלט. למשל דוח אחד יראה רק את סה"כ המכירות בארץ, אחר את המכירות באיזורים 1 ו 2 וכך הלאה.

באיור 2.08 הקמנו לוח פלט שמראה את מירב הנתונים מכיוון ששמות כל השדות כתובים בתחום הפלט וקראנו לו Out1. בחלק הראשון של תרגיל 2.06 אנו רואים לוח פלט ששולף רק את הנתונים הכלל ארציים (שדה הסה"כ) וזאת מכיוון שלא כתבנו את שמות השדות של האיזורים בתחום הפלט - Out2.

בחלקו השני של התרגיל נשלפים כל הנתונים אך סדר הופעת האיזורים בדוח יהיה משמאל לימין בעוד שבתחום הקלט הוא מימין לשמאל.

בחלקו השלישי של התרגיל רואים שילוב של שליפת נתונים חלקית עם עיבוד נתונים. כאן שולפים את פירטי הפריט ואת נתוני איזור מספר 2 לאיזור מספר 3. עמודת הסה"כ כאן נועדה לסכם את שני האיזורים הללו בלבד - מספר שלא נמצא בתחום הקלט. (וזאת משום שבתחום הקלט נחזיק רק נתונים בסיסיים.) בחלק זה של התרגיל רואים כיצד להקים תשתית לחישוב אוטומטי של סה"כ לכל אחת מהעמודות ולכל אחת מהשורות.

כפי שהוסבר בפרק א אפשר להחזיק בראש העמודה משוואה לסה"כ (או אחרת) שתכסה מספר רב של שורות. לכן עם שליפת הרשומות יופיע בראש כל אחת מהעמודות בלוח הסה"כ של אותה עמודה (ראה תרגיל). בתא שנועד לסה"כ הכללי נשמור את נוסחת סך הכל של שני הסה"כ האיזוריים ונגדיר את התא הזה מראש על ידי

/rncLine1

תרגיל 2.06
הקמת תחומי פלט שונים ל Data Query
1. שליפת חלק מהשדות

1		לוח 2.03 - סך הכל מכירות בכל הארץ לפי פריטים				
2						
3						
4						
5				פריט		
6			====	====	====	====
7			תאור		מספר	
8			====	====	====	====
9		סה"כ	צבע	גודל	קט.	רץ
10		====	====	====	===	====
11			סך הכל			
12		====				
13	Out2 ---- >	x5	x6	x7	x8	x9

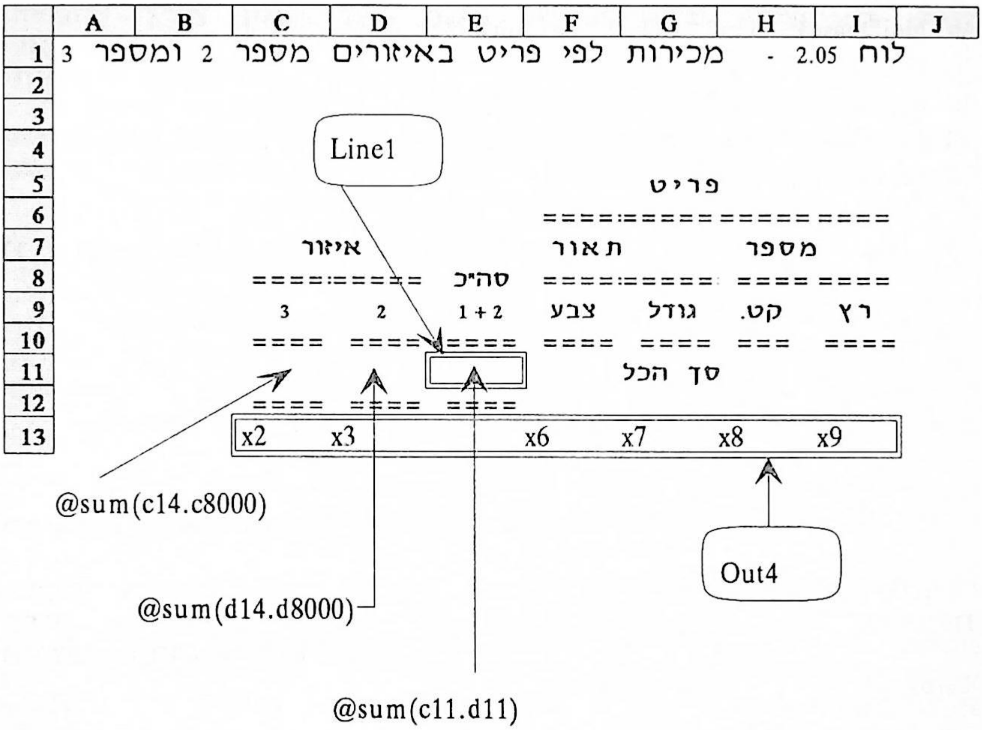
תרגיל 2.06
הקמת תחומי פלט שונים ל Data Query
2. שינוי סדר השדות

1	לוח 2.04 - מכירות לפי פריט ולפי איזורים								
2									
3									
4									
5							פריט		
6						====	====	====	====
7		איזור				תאור		מספר	
8	====	====	====	====		====	====	====	====
9	1	2	3	4	סה"כ	צבע	גודל	קט.	רץ
10	===	====	====	====	====	====	====	===	====
11							סך הכל		
12	===	====	====	====	====				
13	x4	x3	x2	x1	x5	x6	x7	x8	x9

Out3

סדר שדות שונה

תרגיל 2.06
הקמת תחומי פלט שונים ל Data Query
3. שילוב שליפה ועיבוד



ואחרי שליפת הנתונים נבצע Copy של Line1 בעמודה הריקה שנועדה לקבל את הנתון הזה.

עד כה הקדשנו את מרבית הדיון להקמת התשתית. הבה נקים עכשיו את משפטי המקרו שיהפכו את התשתית למערכת שניתן להפעילה. לשם כך נאמר שאם הוקמה תשתית לפי הדוגמה באיור 2.11 הכוללת את המרכיבים הבאים:

לפי איור 2.06	Dbase1
הבינוניים והגדולים	Crt1
הבינוניים הכחולים	Crt2
מכירות לפי פריט ולפי איזורים (איור 2.08)	Out1
מכירות לפי פריט באיזורים מספר 2 ומספר 3 (תרגיל 2.06-3)	Out2

אזי משפטי המקרו להפעלה אוטומטית של כל השאילתות האפשריות הם:

```
\a /dqiDbase1~cCrt1~oOut1~eq{quit}
\b /dqiDbase1~cCrt2~oOut1~eq{quit}
\c /dqiDbase1~cCrt1~oOut2~eq{21}{quit}
\d /dqiDbase1~cCrt2~oOut2~eq{21}{quit}
21 {goto}Line1~{d 2}/cLine1~.{r}{31}{l}~/rv{31}~~{return}
31 {end}{d 2}{end}{u}{return}
```

ובמקום כל שהוא נרשום לעצמנו מקרא:

כל האיזורים:	הבינוניים והגדולים	Alt a
	הבינוניים הכחולים	Alt b
איזורים 2+3:	הבינוניים והגדולים	Alt c
	הבינוניים הכחולים	Alt d

הקם את התשתית הזאת והפעל את המקרו.

משפטי המקרו הללו ניתנים לשיפור ע"י משפט שירות נוסף. כתוב את משפט השירות הזה ושלב אותו בכל אחד מארבעת המשפטים הראשיים שלמעלה.

המשפט הוא: /dqiDbase1~

יש עוד מקרים מסוימים, כגון כשאין בתחום בקלט רשומות תואמות פרופיל, שניתן לטפל בהם בטכניקות שנלמד בפרקים הבאים.

2.3.5 אופציות נוספות ב Data Query

הפקודה Data Query מעמידה לרשות המשתמש מספר אופציות. מתוכן נבחנה עד כה בפירוט האופציה Extract המפעילה את נוהל השליפה של הפקודה. אופציה נוספת המופעלת עם תשתית זהה היא הפקודה Unique. גם פקודה זו שולפת רשומות

> Data Query היא אחת הפעולות המהירות, היעילות והרבגוניות ביותר בגיליון. שווה להשקיע וללמוד.

מתחום קלט אל תחום פלט על פי פרופיל קריטריונים. אבל אם בקלט יש רשומות עם נתונים זהים בכל השדות ששמותיהם מופיעים בתחום הפלט תישלף רק רשומה אחת.

בשתי האופציות האחרות של Data Query אין צורך להגדיר תחום פלט מכיוון שכאן מתייחס המחשב באופן אוטומטי אל תחום הקלט בו-זמנית כאל פלט. האופציה Find מחפשת רשומות על פי פרופיל הקריטריונים, צובעת את הרשומה הראשונה שתואמת את הפרופיל ומאפשרת הנעת הסמן ברשומה שנצבעה משדה לשדה לצורך הקלדת שינויים. הקשה על מקשי החיצים צובעת לפי הסדר את כל הרשומות תואמות הפרופיל שהוגדר.

האופציה Dalete מאפשרת ביטול רשומות על פי פרופיל קריטריונים. הרשומות המסומנות כתואמות לפרופיל יבוטלו בתחום הקלט כאילו ביצענו את הפקודה Worksheet Delete Row על כל אחת מהן בהבדל חשוב אחד. בעוד שביטול שורה בגיליון מבטל את השורה בכל אורכה, דהיינו את כל עמודותיה, ביטול שורה ב Data Query מבטל אותה אך ורק בתחום הקלט.

שימושים מתוחכמים במקרו לשליטה בבסיס נתונים נלמד בפרק הבא. אך לפני שנעבור אליו נפתור תרגיל נוסף ב Data Query.

2.3.6 התאמת בנק ב Data Query

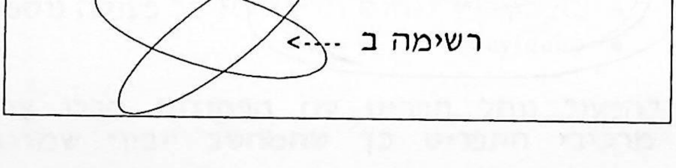
מטרתו של הקטע הנוכחי להדגים דרך חשיבה אפשרי בניתוך בעיה שכיחה למדי והתאמת כלי הגיליון לפתרונה. אם כי התרגיל הזה איננו מפעיל פקודות מקרו אפשר לומר שהוא מפעיל דרך חשיבה חיונים להבנת שפת המקרו.

נתייחס לנושא צעד צעד תוך שימוש בשפה הקרובה ככל האפשר לשפת היום-יום המקובלת כדי שנהיה מציאותיים ככל האפשר.

לפני שתמשיך לקרוא נסה לענות על הגדרת הצרכים הבאה ע"י הפעלה פשוטה של Data Query.

הגדרת הצרכים:

צריך לבצע התאמת בנק.



פתרון

התשתית להתאמת בנק מורכבת משתי רשימות. לדוגמה אפשר לאמר שבכל רשימה שני שדות זהים. בשדה האחד פרטים מזהים של הרשומה כמו מספר צ'ק ותאריך (חוברו כאן לשדה אחד לצורך הדוגמה) ובשני הסכום. כל רשומה היא, נאמר, צ'ק אחר.

רשימה א היא רשימת הצ'קים שערכנו מהספחים שנותרו בפינקס הצ'קים שלנו. רשימה ב קיבלנו מהבנק.

כדי להפיק התאמת בנק צריך לקבל שלוש רשימות כדלקמן:

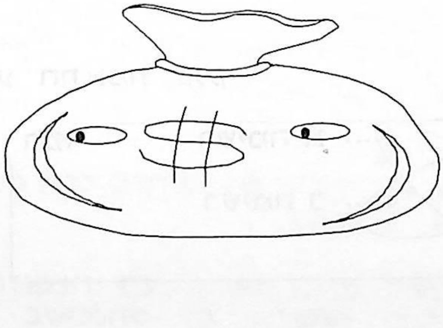
1. צ'קים שרשומים: אצלנו וגם בבנק
2. רק אצלנו
3. רק בבנק

כדי לקבל את הצ'קים שמופיעים בשתי הרשימות נבצע שליפה כשאחת הרשימות מוצבת בתחום הקלט והשניה בתחום הקריטריונים.

כדי לקבל את הצ'קים שמופיעים רק באחת הרשימות ולא בשניה נבצע ביטול כאשר הרשימה המשותפת נמצאת בתחום הקריטריונים והרשימה שלנו (או של הבנק) בתחום הקלט.

ולבסוף צריך להוסיף שדה לרשימה המשותפת ולשלוף אליו נתונים כך שבשדה אחד יהיה זיהוי, בשני סכומים מהבנק ובשלישי סכומים מהרשימה שלנו.

אחרי שמבצעים פעם אחת ידנית את כל הפעולות בהתאמת הבנק (הנ"ל הוא רק חלק מהתהליך כולו) מקימים תשתית ומספר משפטי מקרו להפעלתה. ואז לא נותר אלא לקלוט את הנתונים, להקיש על מקש או שנים ולקבל מיד תדפיס מלא של הפלט המבוקש.



פרק ג: תפריט פקודות "עשה זאת בעצמך"

3.1 מרכיבי תפריט הפקודות

לעיתים קרובות צריך להקים מספר לא קטן של משפטי מקרו כדי לבצע מטלות שונות בגיליון. איך נזכור איזה מקרו מפיק איזו מטלה? דרך אחת היא לרשום על המסך מקרא, כמו במפה, ולפי זה נדע מה להפעיל לפי הצורך. אולם מקרא זה מגושם, לא נוח, מחזיק מקום רב על המסך ולעיתים אף מבלבל.

מקרא, שהוא למעשה תוכן עניינים תוך-גיליוני, אפשר להכין בצורה קלה ונוחה כתפריט פקודות שיראה בדיוק כמו תפריט הפקודות של הגיליון האלקטרוני ויפעל כמוהו. אלא שהוא יפעיל את משפטי המקרו שלנו ואנחנו נקים אותו בעצמנו.

היום למעשה כל תוכנה מאפשרת למשתמש לתת פקודות למחשב באמצעות תפריט. במקום כל שהוא על המסך רואים רשימת פקודות. וכדי לתת את הפקודה מציבים את הסמן עליה ומקישים ENTER. תוכנות רבות מאפשרות מתן הפקודה על ידי הקשת האות הראשונה משמאל (למשל C בשביל Copy) או על ידי הקשה על מספר המסמל את הפקודה (למשל 9 בשביל צא_9). כדי לאפשר את הביצועים הללו צריך תחילה ליצור נוהל תיקשורת. בנוהל זה מצמידים לכל מטלה שלושה דברים:

1. מלת הפעלה (פקודה)
2. תאור
3. משפט מקרו מפעיל

למשל

1

פקודה = ש מ ו ר
תאור = שמור את הקובץ הנוכחי מבלי לצאת ממנו
משפט המקרו = /fs~r{quit}

2

פקודה = כוכבית
תאור = כתוב כוכבית בתא b5
משפט המקרו = {let b5,"*"}~{quit}

3

פקודה = צא
תאור = צא לגיליון לוטוס חדש ללא כל פעולה נוספת
משפט המקרו = /wey{quit}

כדי שאפשר יהיה להפעיל נוהל תפריט עם הפקודות הללו צריך לתעד בגיליון את מרכיבי התפריט כך שהמחשב "יבין" שמדובר

בנוהל תפריט כשיקבל את ההוראה המתאימה.

בוחרים בגיליון מקום שמכיל בדיוק שלוש שורות רצופות הנפרשות על פני מספר עמודות רצופות אך לא יותר משמונה. שלושת התאים בכל אחת מהעמודות הללו מיועדים לתיעוד פקודה אחת. בתא הראשון נרשום את הפקודה, בשני את תאורה, ובשלישי משפט המקרו המפעיל. רוחב העמודה איננו משנה.

מכאן ברור שתפריט אחד יכול להכיל עד שמונה פקודות.

בגיליון אחד אפשר להקים כמה תפריטים שרוצים ולקשר ביניהם.

שטח התיעוד של כל תפריט בגיליון יקבל שם על ידי הפקודה Range Name Create. אין צורך לצבוע את כל שטח התיעוד בעת הגדרת השם - מספיק לצבוע תא אחד בלבד בפינה השמאלית העליונה של שטח התיעוד.

התא בשורה הראשונה מימין לשטח התיעוד חייב להיות ריק. ובתא בשורה הראשונה משמאל לשטח התיעוד נכתוב, לתזכורת, את השם שנתנו לשטח הזה. למען הסדר טוב נותנים לשטחי התיעוד של תפריטים בגיליון את השמות

Menu1 Menu2 Menu3 . . .

וכך הלאה.

איור 3.01

שטח לתיעוד תפריט בגיליון

	A	B	C	D	E	F	G	H	I	J
1		Menu1				שורה לפקודות				ריק
2					שורה לתאורי פקודות					
3				שורה למשפטי מקרו						
4										

למשל תיעוד התפריט לשלושת הפקודות בדוגמה שבתחילת פרק זה יראה בגיליון כך ...

	A	B	C	D	E	F
1		Menu1	שמור	כוכבית	צא	
2			שמור בלי לצאת	כוכבית לתא b5	צא בלי לשמור	
3			/fs~r{quit}	{let b5,"*"}~{quit}	/wey{quit}	
4						

אחרי תיעוד התפריט בגיליון - כולל יצירת שמו - אפשר להפעיל

אחת משתי פקודת מקרו להפעלת נוהל תפריט פקודות בגיליון

{menubranch X} או {menucall X}

הארגומנט X בשתי הפקודות הוא שמו של שטח התיעוד. בדרך כלל נקרא למשפט שמפעיל תפריט בשם \z מכיוון שצמד המקשים Alt z סמוכים זה לזה בצד שמאל ולמטה במקלדת ונוח להפעילם באצבעות יד אחת. למשל
\z {menubranch Menu1}

עכשיו כשנקיש Alt z תראה רשימת הפקודות משורת התיעוד הראשונה בשורת הדו-שיח השניה מעל הסרגל העליון והפקודה הראשונה משמאל (שמור בדוגמה זו) תואר על ידי סמן בוהק. בשורה שמתחת לה (שורת הדו-שיח השלישית מלמעלה מעל לסרגל העליון) יראה ההסבר שתועד בגין פקודה זו בגיליון. הנעת הסמן על פני הפקודות שבתפריט מחליפה גם את ההסבר בהתאם לפקודה עליה מצביעים באותו הרגע. הנעת הסמן מתבצעת לפי אותם הכללים הקובעים את תנועתו בתפריט הפקודות הרגיל של הגיליון.

כאשר מצביעים על פקודה בתפריט ומקישים ENTER קורא המחשב את משפט המקרו המתועד בגיליון בגין פקודה זו ומבצע אותו. כמו בתפריט המקורי של הגיליון אפשר להפעיל את הפקודה ע"י הקשת האות הראשונה משמאל. בעברית נוסיף מספר לפקודה (ראה דוגמה) ואפשר להקיש את מספר הפקודה ולחסוך את הולכת הסמן בתפריט והקשת ENTER.

צא_3 כוכבית_2 שמור_1

יכל להיות שהתפריט הופעל אבל רוצים לצאת ממנו מבלי לבצע אף פקודה. לשם כך נקיש Esc והמחשב יצא מנוהל תפריט הפקודות. וכאן ההבדל בין שתי פקודות המקרו להפעלת התפריט.

כאשר נצא מנוהל שהופעל על ידי {... menubranch} יתנהג המחשב כאילו הסתיים משפט המקרו ב {quit} ופעילות המקרו תעצר. אולם אם נצא מנוהל שהופעל על ידי {menucall...} יצא המחשב מנוהל התפריט וימשיך לקרוא את התווים שמימין בדומה לנוהל משפט שירות כפי שלמדנו בפרקים הקודמים. כדי לעמוד על חשיבותו של נוהל תפריט כמשפט שירות נעשה אתנחתא ונלמד משפט מקרו נוסף

{branch X}

פקודת מקרו זו נכתוב אך ורק בסיום משפט מקרו. מלת ההפעלה branch מורה למחשב לעבור אל (להסתעף) הכתובת שבארגומנט

x ולקרוא את התווים שם כמשפט מקרו.

כדי לרענן את הזיכרון, יחד עם פקודה זו יש שלוש דרכים לסיום משפט מקרו

עצור - המשפט הסתיים ואין לו המשך {quit}
חזרה למשפט ראשי בסיום משפט שירות {return}
הסתעף אל משפט המקרו שכתוב ב x ואל תחזור לכאן {branch X}

הבה נבחן את משפט המקרו הבא
\z {menucall Menu1}{branch \z}

משפט זה מפעיל תפריט פקודות שתיעודו נמצא ב Menu1 בגיליון. מכיוון שהשתמשנו בפקודה {menucall ..} ולא בפקודה {menubranch ...} אזי אם נקיש Esc יצא המחשב מנוהל תפריט, וימשיך לקרוא את הפקודה שמימין. אלא שפקודה זו למעשה **חוזרת ומפעילה את אותו התפריט מחדש.**

בכך יצרנו לולאה אין סופית שנועלת את המשתמש על התפריט ומונעת ממנו לשוטט על פני הגיליון באופן לא מבוקר. כדי לצאת מהלולאה מקישים על צמד המקשים Ctrl Break.

העתק לגיליון את התיעוד מאיור 3.01 והפעל עליו את פקודת התפריט בשתי האופציות שלמדנו.

3.2 אירגון הגיליון לתפריטים

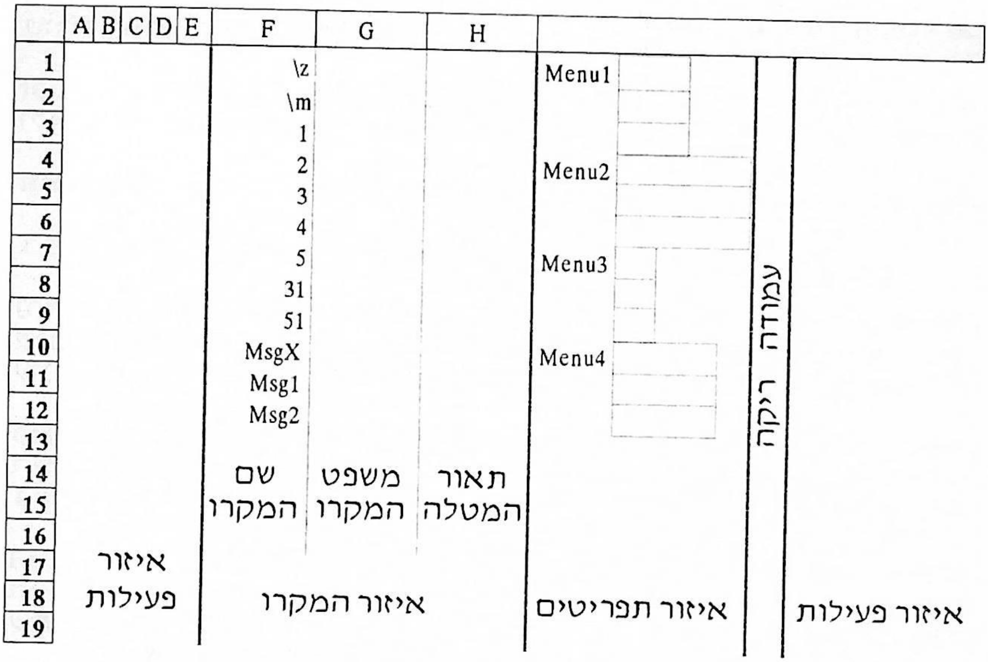
כמו יתר הדברים בכתיבת מקרו גם התפריט הוא חלק ממערך מתוכנן. לשם נוחיות ויעילות ההפעלה מחלקים את הגיליון לשלושה איזורי עבודה ראשיים - איזור פעילות, איזור מקרו ואיזור תפריטים (ראה איור 3.02).

איזור הפעילות יכל להיות מפוצל במקומות שונים בגיליון ובדרך כלל תתחיל הפעילות במסך המראה את חלקו השמאלי העליון של הגיליון ותמשיך, אם יש צורך בכך, בעמודות שמימין לאיזור התפריטים.

איזור המקרו כולל שלוש עמודות. בעמודה הראשונה נכתוב, למטרת תיעוד, את שם המקרו. בעמודה השניה נכתוב את משפט המקרו עצמו ובעמודה השלישית נכתוב מספר מילים לתיעוד תפוקת משפט המקרו.

בעמודה שמימין לאיזור המקרו יתחיל איזור תפריטי הפקודות. בעמודתו הראשונה נתעד את שם התפריט ובעמודות שמימין לה נתעד

איור 3.02
אירגון הגיליון לתפריט פקודות



את התפריט עצמו. מימין לאיזור התפריטים תהיה תמיד עמודה אחת ריקה ואחריה אפשר להמשיך את איזור הפעילות. איזור התפריטים מתפרש על עד תשע עמודות. הראשונה משמאל לשמות התפריטים והיתר לתפריטים עצמם.

כל אחד מהאיזורים הללו משתרע על פני עמודות לכל אורכו של הגיליון מלמעלה ועד למטה ללא כל הצטלבות ביניהם.

כדי שאפשר יהיה לארגן כך את הגיליון לא נכתוב את משפט המקרו עצמו בשורה השלישית של תיעוד התפריט אלא אם יש בה מספר תווים מועט. את משפט המקרו נכתוב באיזור המקרו ובשטח התיעוד נכתוב את פקודת ההסתעפות {branch ... }.

F	G	H	I	J	K
\s	/fs~r{quit}	Menu1	שמור_1	כוכבית_2	צא_3
1	{let b5,"*"}~{quit}		שמור בלי לצאת	כוכבית לתא b5	צא בלי לשמור
2	/wey{quit}		{branch \s}	{branch 1}	{branch 2}

איזור מקרו — איזור תפריטים

באיור הותאם רוחב העמודות לטקסט הכתוב בהן. למעשה, גם בעמודה של משפטי המקרו וגם בעמודות התפריט כל רוחב טור יהיה בסדר, אפילו 1. אין צורך להפריד בין התפריטים על ידי שורת רווח.

זכור - כשנותנים מספר כשם של מקרו הוא נכתב כ LABEL ולא כ VALUE.

ארגן מחדש את התפריט שכתבת בתרגיל הקודם לפי הדוגמאות שלמעלה.

3.3 עץ תפריטים

נרחיב עתה את הדיון בנושא לפתרון בעיות מורכבות יותר. הפתרונות שנבחן יענו על השאלות - מה עושים כשיש יותר משמונה פקודות וכיצד עונים על מצב בו צריך לבחור אופציה מתוך מבחר נתון אחרי הקשת פקודה.

כאשר צריך להפעיל יותר משמונה פקודות על ידי תפריט נקרא לפקודה השמינית בתפריט בשם - עוד (More) - ומשפט המקרו שלה יפעיל תפריט נוסף. בצורה זו נוכל להפעיל מספר רב של פקודות.

הנסיון מראה שברוב המקרים מטלות מתפצלות על ידי אופציות שונות. הפקודה צייר יכולה להתפצל למספר פקודות - משולש, רבוע, וכו. הפקודה שמור יכולה להתפצל לשתיים: המשך או בטל. וכמוה גם הפקודה מחק ואחרות. שליטה במערך אופציות מתבצעת על ידי הקמת עץ תפריטים.

איור 3.03
לוח תכנון תפריט ראשי - Menu1
להפעלה ע"י \z {menucall Menu1}{branch \z}

מלת הפעלה (פקודה)	תאור המטלה	משפט המקרו
שלוף_1	שלוף פריטים לפי קריטריונים שונים	{menucall Menu2}
חפש_2	חפש ומצא פריטים לפי קריטריונים	{menucall Menu3}
מיין_3	מיין את הדוח לפי דרישות שונות	{menucall Menu4}
שורה_4	הוסף רשומה לבסיס הנתונים	{branch 1}
הדפס_5	הדפס את דוח הפריטים שנשלפו	{branch 9}
מסך_6	חזרה למסך הפתיחה ללא כל פעולה נוספת	{branch 10}
צא_7	צא לגיליון לוטוס חדש	{branch \q}
שמור_8	שמור את השינויים מבלי לצאת מהקובץ	{menucall Menu5}

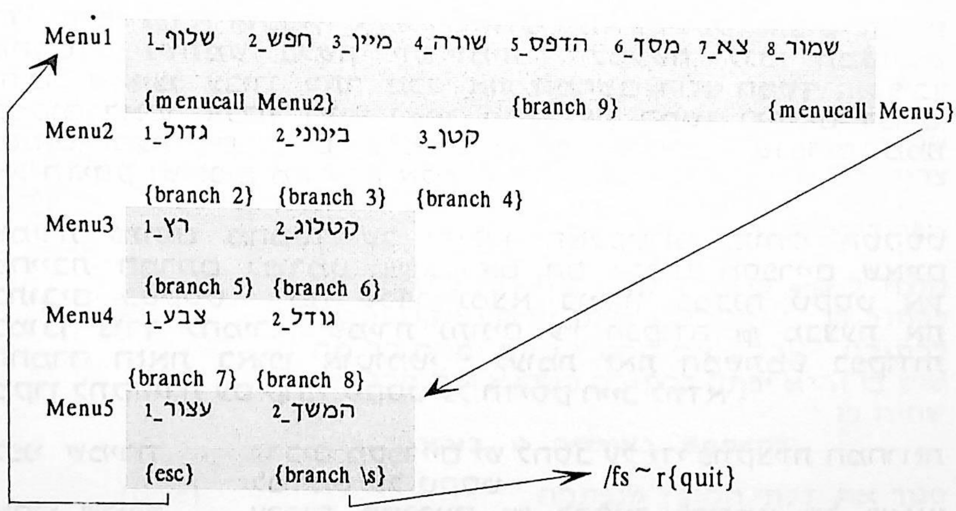
נאמר שהקמנו בסיס נתונים בגיליון כדוגמת בסיס הנתונים בפרק הקודם. בבסיס נתונים זה אפשר לבצע מספר רב ביותר של מטלות. נתאר לעצמנו שיש מספר מסוים של מטלות שמבצעים אותן מספר רב של פעמים ולשם כך הקמנו את משפטי המקרו הבאים.

\z	{menucall Menu1}{branch \z}
\s	שמור בלי לצאת מהגיליון
\q	צא לגיליון חדש בלי לשמור
1	הוספת שורה לבסיס הנתונים
2	שליפת הפריטים הגדולים
3	שליפת הפריטים הבינוניים
4	שליפת הפריטים הקטנים
5	חפש רשומה לפי מספר רץ
6	חפש רשומה לפי מספר קטלוג
7	מיין את הדוח לפי צבע הפריט
8	מיין את הדוח לפי גודל הפריט
9	הדפס את הדוח
10	חזרה למסך הפתיחה

כאשר כל אחד מהמשפטים הללו (פרט ל \q) מסתיים ב {branch \z}.

איור 3.04 מדגים את מבנה עץ התפריטים לדוגמה הנוכחית אשר בה יש הסתעפויות מהתפריט הראשי אל תפריטי משנה. אם צריך אזי גם תפריטי המשנה יכולים להמשיך ולהתפצל עד כמה שרוצים.

איור 3.04
דוגמה לעץ תפריטי פקודות



לכל אחד מהמשפטים 1..10 באיור 3.03 כתוב משפט מקרו כל שהוא. למשל - {d 2}{r 7} וסיים כל אחד מהם ב {branch \z}. העתק את יתר המשפטים לגיליון. הקם עץ תפריטים שלם לפי הדוגמה באיור 3.04 והפעל אותם. איזה שימוש מיוחד נעשה כאן בתפריט ע"י Menu5?

פרק ד: ASCII

4.1 מה זה ASCII

ראשי תיבות

American Standard Code for Information Interchange

סימול אחיד זה של תווים וסימני טקסט מאפשר איכסון מידע בצורה יעילה וחסכונית. באופן מיוחד חשובה שיטה זו מכיוון שהיא מאפשרת העברת מידע שנוצר באמצעות תוכנות שונות בין מחשבים ובין תוכנות.

קובץ ASCII הוא קובץ טקסט המשתמש בסמלים הללו לייצוג תווים וסמלי טקסט.

בגיליון האלקטרוני יש שני נוהלים לייצירת וניהול קובצי טקסט. הנוהל האחד מתבצע על ידי פקודות התפריט Print File ו File Import. הנוהל השני מתבצע על ידי סידרה של פקודות מקרו. שני הנוהלים הללו עוסקים בתיקשורת בין הגיליון לבין הדיסק (או הדיסקט). נושא זה נידון בהרחבה בסיפרי "מקרו למתקדמים בלוטוס" בהוצאת הוד-עמי אך מן הראוי להקדיש גם כאן מספר מילים לנושא.

כדי שגם קורא שעדיין איננו "מקצועי" מאוד יוכל להפיק תועלת מההסבר נשתמש בלשון יום-יומית. למעשה, לגבי המשתמש הדבר שמייצג עבורו יותר מכל את המחשב הוא המסך. לפיכך במקום לאמר זיכרון דינמי נאמר מסך. על המסך כותבים דברים, מזיזים, מעתיקים, ממנו שומרים ואליו שולפים. שומרים ממנו אל הדיסק ושולפים מהדיסק אליו.

שמירת נתונים מהמסך של הגיליון האלקטרוני בנוהלי הטקסט מחייבת המרתם לפורמט LABEL אם הם ערכים מספריים שאינם כתובים כטקסט. מה שכבר נמצא בגיליון במבנה טקסט אין כמובן צורך להמיר. שמירת נתונים ע"י הפקודה /pf מבצעת את ההמרה הזאת באופן אוטומטי. לעומת זאת המשתמש בפקודות מקרו לתקשורת עם קובץ טקסט על הדיסק חייב לוודא -

לפני שמירה - ערכים מספריים יש להסב על ידי פונקציות המחרוזת לפורמט של טקסט

אחרי שליפה - ערכים מספריים יש להחזיר לפורמט של VALUE על ידי פונקציות המחרוזת

אפשר לאמר שמי שאיננו מפעיל את הגיליון בנוהלי ASCII

מנצל חלק קטן מאוד מהאופציות שיש בו. בפרק זה נתייחס רק לנוהל ASCII על ידי פקודות מקרו וגם זה נעשה על קצה המזלג.

4.2 מקרו ל ASCII

בעבודה עם קובץ טקסט, דהיינו בתיקשורת בין המסך לבין קובץ טקסט על הדיסק יש מספר סוגי פעולות.

1. פתיחת קובץ על הדיסק (כמו שפותחים דלת לחדר)
2. כתיבה מהמסך אל תוך הקובץ שנפתח
3. קריאה מתוך הקובץ שנפתח (שליפה אל המסך)
4. סגירת הקובץ שנפתח (כמו שסוגרים דלת)

קבוצת פקודות המקרו לנוהל עבודה עם ASCII עונה על סידרה של מצבים אפשריים שונים ונבחר להשתמש בפקודה המתאימה בהתאם לתמונת המצב ברגע ההפעלה.

4.2.1 פתיחה לשם כתיבה

```
\z        {open FileName1,w}
          {close}
          {quit}
FileName1 c:\Data1.Dat{return}
```

לפנינו פיסקת מקרו ל ASCII המשלבת מספר פקודות מקרו במערך מודולרי.

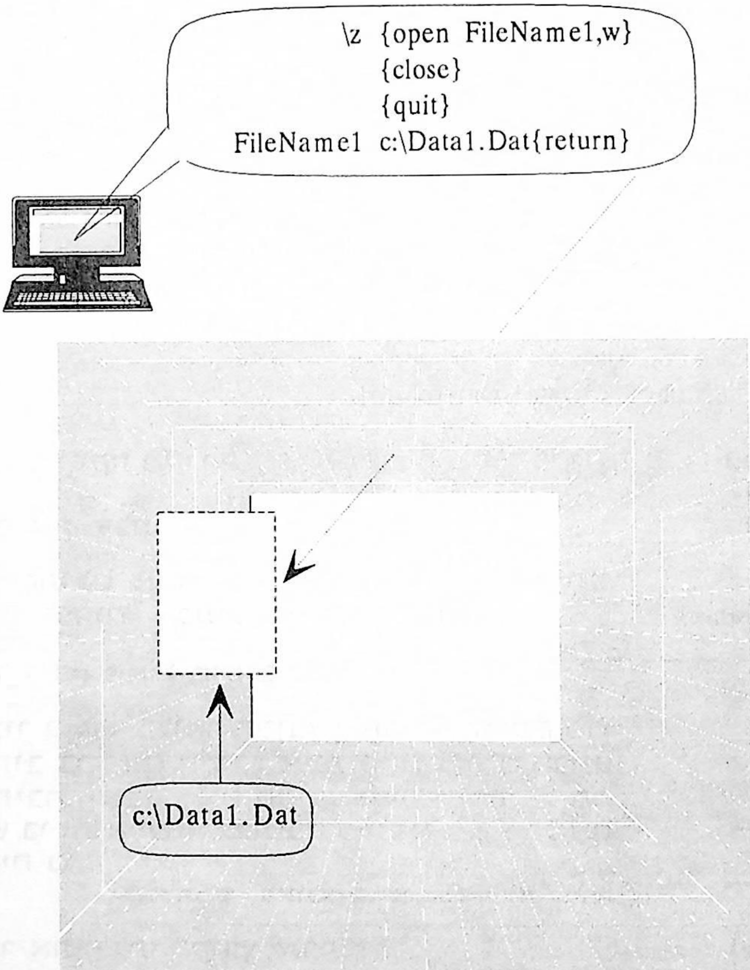
פתח את "דלתו" של קובץ טקסט	{open
שם התחום אשר בו כתוב שם הקובץ שאותו צריך לפתוח - כתוב כמשפט שירות	FileName1
מפריד בין הארגומנטים	,
קיצור ל write, כלומר הקובץ נפתח על הדיסק כדי לכתוב בו. אם הקובץ איננו קיים - יוקם ויפתח לכתיבה. אבל - אם הקובץ קיים ימחק כל מה שיש בו והוא יפתח לכתיבה כשהוא ריק מכל מה שהיה בו. **להשתמש באופציה זו בזהירות רבה**	w}
סגור את "דלת" הקובץ שנפתחה	{close}
סוף פיסקת המקרו	{quit}

ואפשר גם לכתוב

```
{open "c:\Data1.Dat",w}
```

ננסה להסביר את התהליך על ידי משל. נאמר שיש כרטסת שמות. והכרטיסים מונחים במגירה לפי סדר האלף בית. אם נשתמש באופציה w (write) ואין כרטיס לאדם ששמו רשום ב FileName1 לוקחים כרטיס חדש, רושמים בו את השם ושמים אותו במגירה. אבל אם יש כבר כרטיס לאדם הזה ובכרטיס כתובים דברים ומשתמשים באופציה w אזי משמידים את הקיים ושמים במקומו כרטיס חדש.

איור 4.01
ה מ ס ך ו ה ד י ס ק



אפשר לפתוח קובץ בכל שם שרוצים ובכל סיומת אבל את המחיצה (directory) צריך להקים מראש אחרת לא יוקם הקובץ שרצינו.

באופציה הזאת (write) משתמשים אך ורק כשרוצים להקים קובץ טקסט בפעם הראשונה או כאשר רוצים להחליף את כל תחולתו של קובץ קיים בטקסט חדש.

אחרי שהוקם הקובץ ונפתח לכתיבה (כאילו לקחנו תיק חדש, כתבנו עליו שם ופתחנו אותו) אפשר לכתוב בו מיד או אפשר לסגור ולשים במגירה. יש לזכור שלעולם סוגרים את הקובץ עם סיום העבודה בו. בדוגמה שלמעלה פשוט הקמנו קובץ, פתחנו אותו וסגרנוהו מבלי לכתוב בו דבר.

אך מה עושים כשקיים כבר קובץ טקסט ורוצים לכתוב אליו או לקרוא (לשלוף) ממנו חלק כל שהוא מבלי למחוק את כל היתר. לשם כך במקום להשתמש באופציה w (write) נשתמש באופציה m (modify).

```
          \z {open FileName1,m}
             {close}
             {quit}
FileName1 c:\Data1.Dat{return}
```

אם הקובץ קיים הוא יפתח מבלי שתוכנו הנוכחי ימחק ואפשר לשלוף חלקים ממנו או לכתוב אליו מה שרוצים היכן שרוצים מבלי למחוק או לפגוע ביתר הטקסט בקובץ.

מה יקרא אם הקובץ איננו קיים.

ראינו שבאופציה w אם הקובץ איננו קיים הוא יוקם. אבל באופציה m (modify) אם הקובץ איננו קיים הוא לא יוקם. המחשב ימשיך לקרוא את פקודות המקרו שמימין לפקודת ה open ובאותה שורה ואחר כך ימשיך לקרוא את הפקודות בשורות הבאות באותה עמודה.

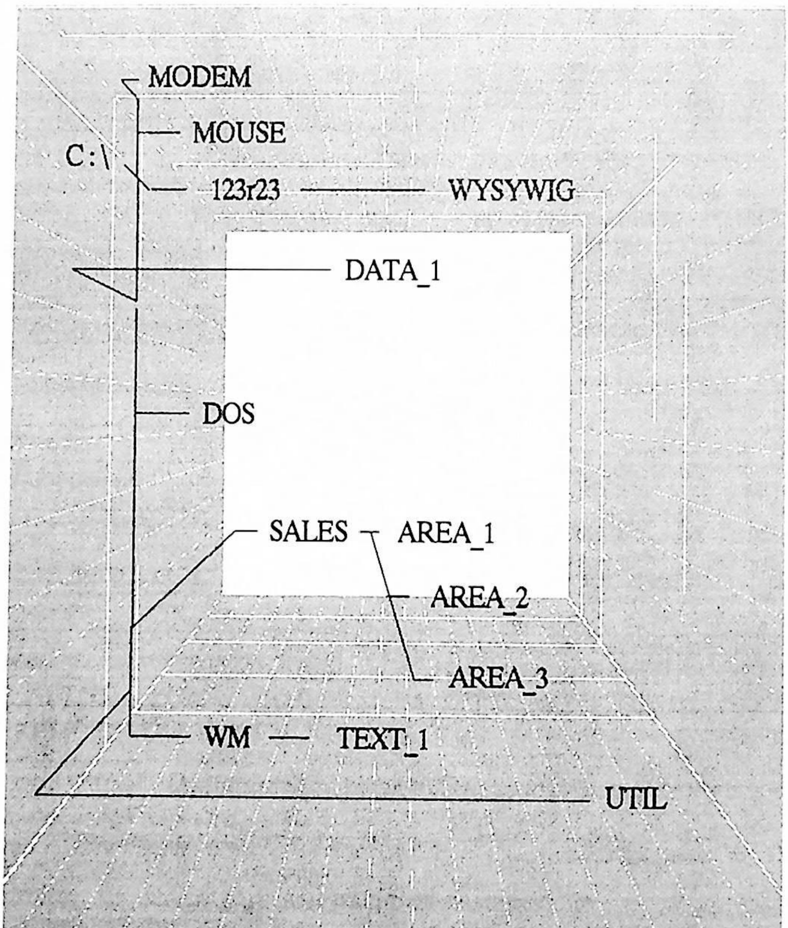
נבחן דוגמה אחת מרבות לשימוש בנוהל שהוסבר עד כאן.

נאמר שיש קובץ טקסט ושומרים אותו בדיסקט. כדי לשלוף ממנו נתונים או לעדכנו צריך להכניס את הדיסקט בכונן A ללא מדבקת ההגנה. משפט המקרו הבא בודק את הדברים הללו ומודיע לנו אם איננו מצליח למצוא את הקובץ.

```
        \z {open FileName1,m}{branch 1}
           {close}
           {quit}
         1 {indicate Msg1}{quit}
 FileName1 a:\Data1.Dat{return}
      Msg1 הדיסקט מוגן או הקובץ לא נמצא
```

מספר פקודות המקרו לעבודה עם ASCII איננו רב אבל כדי ללמוד להפעילן צריך יריעה רחבה ביותר ונסיון כל שהוא בעבודה עם מקרו. הסיבה לכך היא שפעילות זו משולבת בנושא של השליטה והבקרה בדיסק באמצעות הגיליון האלקטרוני שאיננו נדון בספר זה.

איור 4.02
מרחב הדיסק



פרק ה: לולאות

5.1 תכונות הלולאה

הלולאה היא אחת מצורות ההתנהגות השכיחות ביותר בכל שנפנה. אוטובוס מגיע לתחנה אוסף ומוריד נוסעים וממשיך בדרכו. הנוסע, מנקודת הראות שלו, עובר על פני מספר תחנות ויורד במחוז חפצו. לא כך הוא הדבר מנקודת ראותו של הנהג. נהג האוטובוס ימשיך בדרכו ולאחר זמן מה יחזור ויגיע אל אותה תחנה. כלומר זרם של נוסעים הועבר ממוצא אל יעד באמצעות האוטובוס בעוד שהאוטובוס שייצר את הזרם הזה נע בתנועה מעגלית חוזרת שאין לה לכאורה לא מוצא ולא יעד.

במחצבות האבן נטחנים סלעים לחצץ. חגורות שינוע מעבירות אבנים גדולות מערימה אחת אל מגרסה הטוחנת אותן לגודל הנדרש. כף אחת בחגורת השינוע נטענת בנקודה אחת, נעה עמוסה עד לנקודה שניה, פורקת את מטענה וחוזרת ריקה וחוזר חלילה- פעם אחרי פעם אחרי פעם ...

המדרגות הנעות עוברות והולכות ללא הפסקה, פעם אחרי פעם. מעבירות אנשים ומשא מקומה אחת לשניה וחוזרות ריקות.

אם תרצו אפשר לזהות לולאות כמעט, אם לא בכל, סוג של פעילות בטבע. ובודאי במחשב. בעבודה עם הגיליון האלקטרוני אפשר לזהות מספר סוגים של פעילויות מחזוריות כאלה - לולאת התניה, לולאה אינטראקטיבית אין סופית ולולאה סידרתית.

5.2.1 התניה עם {if ... }

את המבנה הלוגי של התניות אפשר להדגים במשפט ...

אם A אז B, אחרת C

בפונקצית ההתניה של הגיליון האלקטרוני יכתב משפט זה ...

```
@if(A,B,C)
```

לעומת זאת במקרו מבנה משפט ההתניה הוא ...

```
{if   A}B
C
```

ובמילים - אם הארגומנט A נכון אזי ימשיך המחשב לקרוא את פקודות המקרו באותה שורה מימין לפקודת ההתניה, אחרת יקרא המחשב את פקודות המקרו הכתובות בשורה הבאה.

תרגיל 5.01
לולאה עם {if ... }

	A	B	C	D	E	F	G	H	I	J
1	דוד	\z {restart}{menucall Menu1}{branch \z}								
2		\b {home}{d 19}{quit}								
3	חנה	1 {goto}Data1~								
4	דן	2 {if @cellpointer("contents"=1)}{branch \b}								
5		{if @cellpointer("contents"="")}/cPic1~~{d}{branch 2}								
6		{if @cellpointer("contents"=Pic1)}/re~~{d}{branch 2}								
7		Pic1								
8	גלי									
9										
10	עופר									
11										
12	מרים									
13										
14	1									
15										
16										
17										
18										
19										
20										

Data1

	E	F	G	H	I	J
1	Menu1	החלף_1	לגיליון_2			
2						
3		{branch 1}	{branch \b}			

בתרגיל 5.01 רשומים שמות בתחום ובין השמות יש שורות ריקות. רוצים שתי אופציות למבנה התחום. באופציה האחת בשורות שבין השמות יראו חמש נקודות ובאופציה השניה יהיו השורות הללו ריקות (ממש ריקות).

העתק את התרגיל לגיליון ובקש הפעלה על ידי התפריט. הפעל מספר פעמים כדי לראות את השינויים המתחוללים. חזור והפעל בהפעלה רגילה ובשלבים.

איך "יודע" המחשב מתי לסיים.

הסיפרה 1 בשורה האחרונה של Data1

חבר משפט מקרו שיבטל את השורות הריקות. ממה צריך להזהר כאן ואיך נפתור את הבעיה?

לקראת מטלה אפשרית של ביטול שורות יש למקם את התחום כך שלא יבוטלו דברים אחרים בגיליון באותה שורה. כמו כן צריך לכתוב נוהל מיוחד לביטול השורה הראשונה או האחרונה בתחום שאם לא כן ישתבשו הקואורדינטות של שם התחום.

5.2.2 {restart} לביטול אפקט הקינון (nesting)

ראינו בפרקים הקודמים שיש שתי דרכים לקישור בין משפטי מקרו - נוהל התפצלות על ידי הפעלת הפקודה { ... branch} ונוהל שירות על ידי קריאה למשפט שירות וסיומו בפקודה {return}. כשקושרים שרשרת של משפטי מקרו עשוי להווצר מצב שכתוצאה ממנו יצטרך המחשב, ברגע מסוים, לחזור על צעדים בסדר הפוך. בדומה למי שנכנס למערה סבוכה וחשוכה וחבל בידו וכשמבקש לצאת חוזר בעזרת החבל. מצב כזה נוצר כשקוראים למשפט שירות מתוך משפט שירות מתוך משפט שירות ... וכשמקימים עצי תפריט.

במקרים כאלה יוצר המחשב מעין "חבל". רשימה שמאפשרת חזרה למשפט המקור. רשימה זו נמצאת במעין "מחסנית" (stalk) במחשב ובדרך כלל איננה מפריעה לפעילות השוטפת. לכל קריאה למשפט שירות (חוליה) בשרשרת הזאת קוראים רמת קינון (nesting level) ומספר רמות הקינון לאיכסון תלויה במחשב ובגירסת הגיליון.

לפעמים אנו עלולים להמצא, בבלי דעת, במצב שבו המחשב נצרך למחסנית אך זו מלאה. במקרה כזה נקבל את הודעת ה ERROR - Too many nesting levels. מאחר ולא תמיד דרוש נוהל זה רצוי מפעם לפעם לנקות את המחסנית - ביחוד כשמפעילים לולאות.

פקודת המקרו שמנקה את המחסנית מכל מה שיש בה היא

{ r e s t a r t }

ואיננה מקבלת ארגומנטים. את הפקודה הזאת נכון לכתוב אך ורק בתחילתו של מקרו שמגיעים אליו על ידי הפקודה {branch}. שכן אחרי פקודה זו מתחילה תמיד שרשרת פקודות חדשה. לעומת זאת אין להפעיל פקודה זו כחלק משרשרת פקודות מקרו שהתחילה על ידי קריאה למשפט שירות או בנוהל לולאה ללא התפצלות על ידי {branch}.

5.2.3 ארגומנטים ב { ... if}

עד כאן ראינו את הצורה הפשוטה ביותר של לולאת התניה כאשר המקרו { ... if} מקבל כל ארגומנט שנרצה. דוגמאות חשובות לארגומנטים במקרו הזה ...

פונקצית התניה בתוך מקרו התניה {if @if(a,b,c)}
כאשר a ו b וכולים להיות מספרים או טקסט {if a<>b}
כשב Test1 יש משוואה חיה הבוחנת התרחשויות בגיליון {if Test1=0}

5.2.4 תרגיל חשיבה ופיתוח

עד כה ראינו איך כתיבת מקרו חובקת מרכיבים שונים בגיליון בנוסף לפקודות המקרו הספציפיות. למכלול המשלב חיבור מקרו עם מבנים בגיליון קראנו - אינטרה מקרו. כאן מוצגת לפני התלמיד בעיה של ממש שפתרונה יעזור בקטע חשוב בעיבוד נתונים תוך גיליוני. פתרון מופיע בסופו של פרק זה.

נתון גיליון אלקטרוני ובו לוח עם נתונים. למשל, כל שורה היא נקודת יישוב בארץ ובעמודות יש פרטים כמו שם היישוב, מספר התושבים, מספר בתי הספר או כל מה שתרצו.

הלוח ממוין מראש לפי סדר הא״ב של השמות.

רוצים להוסיף עוד שורה בשביל יישוב חדש.

רבים פשוט מוסיפים שורה בסוף וממיינים מחדש. אלא שמיון של לוחות גדולים לוקח זמן רב ויכולים להווצר שיבושים במשוואות ובהגדרות התחומים.

נוהל חלופי הוא "לרוץ" על פני השמות, לגלות את המקום המדויק בו צריך השם החדש להופיע ולהכניס שורה ע"י /wir~.

כתוב את חיבור האינטרה מקרו לביצוע המטלה הזאת כאשר בסיס הנתונים הנתון הוא בן עמודה אחת ומספר שורות

Data1 --- >

דוד
חיים
טובה
ישראל
כוכבית
מרים
נועם
נעמה
פנינה
תמר

5.2.5 התניה עם {open ... }

בפרק הקודם למדנו כי נוהל {open ... } לעבודה עם ASCII מותאם גם לפעול כהתניה. נביא כאן דוגמה נוספת ליישום המקרו הזה כלולאת התניה.

איור 5.01
הגנה מהעתקה לא חוקית

	A	B	C	D	E	F	G	H	I
1		\0 {open FileName1,m}{branch 1}							
2		{close}							
3		{branch 2}							
4		1 {beep 1}{beep 1}{beep 1}{beep 1}{branch 1}							
5		2 {home}{quit}							
6	FileName1	c:\mycode.pro{return}							
7									

אחת הדרכים להגן מפני העתקה לא חוקית של גיליון היא הקמת נוהל אוטומטי שבעזרתו "יודע" הגיליון - ברגע שנשלף אל המסך - אם הוא נמצא במחשב הבית שלו או במחשב זר. לשם כך מבצעים את הפעולות הבאות.

במחשב הבית מבצעים את הפעולה

```
{open FileName1,w}
{close}
```

כאשר ב FileName1 כותבים איזה שם קובץ שרוצים. לאחר מכן עוברים ל DOS ומבצעים (רצוי בעזרת תוכנת שירות) הגנה על הקובץ הזה כך שיהיה גם קובץ חבוי (hidden) וגם לקריאה בלבד. נוהל פתיחת הקובץ מבטיח שיהיה זה קובץ ריק. אח"כ חוזרים לגיליון וכותבים את משפט המקרו שבאיור 5.01.

שם המקרו יהיה \0 וכתוצאה מכך יופעל אוטומטית מיד עם הופעת הקובץ על המסך.

כתוב את סדר הפעולות אם הקובץ מופעל ממחשב הבית.

כתוב את סדר הפעולות אם הקובץ מופעל ממחשב זר.

הקם את האינטרה מקרו הזה והפעילו.

איך מפסיקים פעולה של לולאה אין סופית?

צמד המקשים Ctrl Break

> המקרו {beep x} מפיק ציפצוף
> הארגומנט x יכל להיות 1 2 3 או 4
> והוא קובע את טון הציפצוף

מה ההבדלים בין ההתניה עם { ... if} לבין ההתניה עם { ... open}?

1. באחד (איזה?) התלות היא באירוע ספציפי ובשני התלות היא בארגומנט שאפשר לכתוב בו מה שרוצים.

2. באחד (איזה?) אם התנאי נכון יבוצע המקרו שכתוב מימין באותה שורה ואילו בשני אם התנאי לא נכון יבוצע המקרו שכתוב מימין באותה שורה

5.2.6 התניה עם { ... onerror}

כל פעילות אנושית משולבת בטעויות וכך גם השילוב אדם-מחשב. החוכמה היא להכיר מראש בעובדה הזאת ולהתכונן לטעויות. וכדי להיות מוכן צריך כלים שיאפשרו - גילוי ההתרחשות של טעות, זיהוי מקור הטעות, ותגובה מתאימה. טעות נפוצה בעבודה עם מחשבים נוצרת כאשר מסיבה כל שהיא המדפסת לא פועלת. טעות נפוצה אחרת היא כאשר מבקשים לבצע פעולה בכונן A והכונן איננו מוכן. פקודת המקרו on error מטפלת בטעויות כאלו.

{onerror x,y}

אם במשפט מקרו צופים אפשרות של טעות משלבים במשפט, לפני המקרו שעשוי לגרום טעות, את פקודת המקרו הזאת. למשל

{onerror WW1,1}/fsX{esc}a:\1~{quit}

לפני שמירת הקובץ בכונן A. לכל ERROR מוגדרת מראש הודעה שמופיעה על המסך כשהטעות מתרחשת. הארגומנט הראשון (לא חובה) מציין כתובת בגיליון (כאן WW1) שלתוכה תיקלט ההודעה. משתמשים בארגומנט הזה כאשר רוצים לידע את המפעיל באשר למה שקרה. כאשר סוג הטעות לא צפוי מראש אפשר להשתמש בהודעה שנקלטה לייצור התפצלות מותנית אל מקרו שיטפל במצב החדש שנוצר. בדוגמה הנוכחית ההודעה הצפויה היא

Disk drive not ready

הארגומנט השני הוא שם תחום של משפט מקרו שמגיב על ההתרחשות של טעות והוא ארגומנט חובה. אם מתרחשת טעות תהיה התפצלות של המקרו בדיוק כאילו השתמשנו בפקודה { ... branch}. למעשה אפשר לומר כי הפקודה on error פועלת כהסתעפות המותנית בהתרחשות של טעות במהלך ביצוע המקרו.

מהרגע שנכתבה הפקודה בתוך משפט המקרו היא תקיפה עד שתתרחש טעות, או עד שתתבטל על ידי פקודת {onerror} ללא ארגומנטים או עד שתופיע במשפט פקודת { ... onerror} חדשה.

תרגיל 5.02
הסתעפות מותנית בהתרחשות של ERROR

	A	B	C	D	E	F	G	H	I
1	\z	{onerror Msg1,1}/fsX{esc}{FileName1}~{quit}							
2	1	{let Msg1,Msg2&Msg1}~{indicate Msg1}{?}{indicate}{branch \z}							
3	Msg1								
4	Msg2	ENTER בדוק את כונן A ואח"כ הקש ==>							
5	FileName1	a:\1.wk1{return}							

הקם את האינטרה מקרו מתרגיל 5.02 בגיליון והפעל אותו באופן רגיל ובשלבים. הפעל פעם אחת כשהכונן פתוח ופעם שניה כשהכונן סגור ומכיל דיסקט.

כתוב את סדר הפעולות כאשר לא מתרחשת טעות.

כתוב את סדר הפעולות כאשר מתרחשת טעות.

הוסף למשפט ציפצוף על ידי משפט שירות בשם Beep1 והפעל.

מצא על ידי ניסוי מדוע /fsX{esc}{FileName1} ולא /fs{FileName1}

כדי שהשמירה תתבצע בין אם הקובץ קיים ובין אם איננו קיים.

5.3.1 לולאת תפריט אין סופית

בפרק ג ראינו כיצד מקימים תפריט פקודות כלולאה

\z {restart}{menucall Menu1}{branch \z}

אינטראקטיבית אין סופית. לולאה זו מנתבת את המשתמש בתוך מערך פקודות מוכן מראש ומה שלא יעשה ימצא את עצמו מבצע אחת מהאופציות או מול תפריט הפקודות. כדי להכנס אוטומטית לנוהל זה עם הופעת הגיליון על המסך נותנים ל \z שם נוסף - \0 וכפי שכבר למדנו שם זה יפעיל את משפט המקרו מיד עם שליפת הקובץ ותפריט הפקודות יופיע באיזור הדו-שיח מוכן להפעלה.

5.3.2 קישור מסך קלט עם תפריט בלולאה אין סופית

בקליטת נתונים אנחנו מכירים שני נוהלים עקריים. בנוהל האחד מוליכים את הסמן, ידנית או על ידי שאילתא, אל התא בו רוצים לכתוב את הנתונים וקולטים אותם שם. חלק גדול ממשתמשי הגיליון נוהגים בנוהל זה שכן הוא הדומה ביותר לכתיבה בעפרון על נייר.

כשמשלבים אוטומציה בעבודה אפשר לעבור לנוהל השני של קליטת נתונים - מסך קלט. בנוהל זה מכינים בגיליון מעין טופס

נוח לקליטת נתונים. המסך מאורגן בצורה נוחה למפעיל והוא איננו צריך להסתבך בתוך לוח מורכב. לאחר הקלדת הנתונים נשתלת הרשומה החדשה במקומה בלוח בהקשת פקודה אחת.

קליטת נתונים בגיליון האלקטרוני באמצעות מסך קלט מאפשרת הקמת מערך ביקורת לוגי שמעביר את הנתונים דרך מסננת הביקורת כדי להבטיח מפני קליטת שגויים בבסיס הנתונים. הגיליון האלקטרוני מאפשר הקמת מגוון בלתי נדלה של ביקורות כאלה - למעשה אפשר "לתפוס" כמעט כל שגיאה שניתן לדמיין מראש.

דרך נוחה לייצירת מסך קלט היא הפעלת הפקודה /ri וכאשר קושרים את הפקודה הזאת עם נוהל תפריט פקודות מקבלים מסך קלט למהדרין.

בספר זה לא נציג את המכלול המלא של מרכיבי מסך הקלט. נתייחס רק לקישור בין נוהל /ri לבין נוהל {menucall ... }.

כדי ללמוד את הקישור הזה הבה נבחן את תרגיל 5.03.

תרגיל 5.03 מציג אינטרה מקרו לקישור /riData1 עם {menucall Menu1}. בחן היטב את התרגיל והקם את האינטרה מקרו הזה בגיליונך. בעת העתקת התרגיל לגיליון יש לשים לב למיקום המרכיבים השונים בעמודות ובשורות.

אחרי שהגדרנו את התחום Data1 וכתבנו בו את הקווים ואת האפסים נבצע unprotect על Data1. כדי להקים את Line1 נציב את הסמן בתא s1, נבצע Range Trnspose מ Data1 ואחר כך Range Name Create Line1 כמתואר בתרגיל.

בדוק שהעתקת את התרגיל במדויק אל גיליונך והקמת את התחומים כפי שלמדנו בפרקים הקודמים. שמור את הגיליון מספר פעמים במהלך העבודה.

עכשיו בצע את הדברים הבאים וענה על השאלות.

הצב את הסמן בתא הראשון של Data1 והקש Alt z. מה התרחש?

הקש על המקש Esc. מה התרחש?

הקלד נתוני דמה ב Data1. בסיום הקש Esc. מה התרחש?

בתפריט הפקודות בקש "קליטה_1". איזה שינויים התרחשו בגיליון והיכן?

> המקש Esc מופעל כאן כנתב. כשרואים את התפריט ורוצים לקלוט נתונים מקישים Esc. כשנמצאים במצב הקלדה ורוצים לקבל את תפריט הפקודות - מקישים Esc.

תרגיל 5.03
קישור מסך קלט עם תפריט בלולאה אין סופית

Data1 → A8:B13 Dbase1 → D8:I10

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8		---	שם משפחה	x1	x2	x3	x4	x5	x6
9		---	פרטי	שמואלי	מרים	דן 87	רחובות	78091	88052
10		---	רחוב ומס	חיים	דוד	הגן 25	חיפה	50557	50467
11		---	יישוב						
12		0	מיקוד						
13		0	טלפון						

	J	K
1	\z	{restart}{menucall Menu1}/riData1~{branch \z}
2	\b	/wtc{home}{goto}Data1~/rtLine1~~/wth{branch 41}
3	1	{goto}Dbase1~{31}{d}/rtData1~~{35}{branch \b}
4	2	/fs~r{branch 41}
5	3	{home}{quit}
6	31	{end}{d 2}{end}{u}{return}
7	35	/rncDbase1~{d}~{return}
8	41	{restart}/riData1~{branch \z}

	M	N	O	P	Q	R	S	T
1	Menu1	קליטה_1	שמור_2	צא_3				
2								
3		{branch 1}	{branch 2}	{branch 3}				

	R	S	T	U	V	W	X
1	Line1	---	---	---	---	0	0
2							
3							

קלוט עוד שלוש או ארבע רשומות. הפעל את המקש Esc ממצבים שונים.

הפעל את כל האופציות בתפריט בשלבים ולא בשלבים.

מדוע הקשה על Esc פועלת בחיבור מקרו זה כפי שהיא פועלת?

בשביל מה צריכים את משפט המקרו 41? מתי נסתעף אליו ומתי נסתעף אל \z? מה ההבדל ביניהם?

מדוע צריך התחום Data1 להיות בפורמט unprotect?

ההקשה על Esc ממצב שהגענו אליו ע"י {menucall ... } משחררת את התפריט והמחשב ממשיך לקרוא תוים במקרו שמימין ל {menucall ... }. הפקודה שנמצאת שם היא ~/riData1 ולפיכך עוברים מיד למצב שבו מתאפשרת הקלדת נתונים אבל ניתן להניע את הסמן רק על פני תאים שהוגדרו מראש unprotect ונמצאים בתחום Data1. במצב זה אין כל אפשרות להניע את הסמן אל מחוץ לתחום Data1. כשנמצאים במצב זה הקשה על Esc מבטלת אותו, אלא שכאן המקרו מסתעף אל \z וחוזר ומפעיל את תפריט הפקודות.

כשמסתעפים אל \z מקבלים תחילה את התפריט ואילו כשמסתעפים אל משפט 41 מקבלים תחילה את מצב ההקלדה.

בצורה זו ייצרנו לולאה אינטראקטיבית אין סופית עם אופציה להניע את הפעילות מכיוון התפריט אל כיוון ההקלדה או להפך.

5.4 לולאות סידרתיות

חלק נכבד מהמטלות בגיליון האלקטרוני ניתן לאוטומציה יעילה בעזרת הלולאה הסידרתית ומשום כך חשיבותה מרובה. את המיומנות בהפעלת הטכניקה הזאת נפתח בהדרגה ומומלץ לא לדלג על שלבים במהלך פרק זה. תחילה נכיר את מרכיבי הלולאה הסידרתית, אח"כ נלמד להקימה בשלבים על ידי פקודת המקרו {let ... } ולבסוף נלמד להפעיל את הלולאה על ידי הפקודה {for ... }. התרגילים בהמשך מותאמים לכל שלב בהסבר ומדורגים לפי רמת המורכבות של המטלה לביצוע עד שמגיעים לבסוף לטכניקה של לולאות מורכבות - הפעלת לולאה מתוך לולאה.

הלולאה יותר משהיא טכניקה היא דרך חשיבה בניתוח בעיות שמובילה לפתרונות אופרטיביים. הבה נבחן שנית את הדרישה מתרגיל 5.01.

בתחום שנקרא Data1 והמשתרע על פני מספר שורות בעמודה אחת רשומים מספר שמות ושורות רווח בינהם. רוצים לכתוב קוים (---) במקום הרווחים.

תרגיל 5.04.1
מרכיבי הלולאה הסידרתית
הולכת סמן בפועל

	A	B	C	D	E	F	G	H	I	J
1			\z	{let Counter1,0} ~ {home}{goto}Data1 ~						
2			1	{let Counter1,Counter1+1} ~ {if Counter1>Stop1}{branch 2}						
3				{if @cellpointer("contents")=""}/cPic1 ~ ~ {d}{branch 1}						
4	דוד			{d}{branch 1}						
5			2	/rlrData1 ~ {home}{goto}Data1 ~ {quit}						
6	חנה		Counter1	0						
7	דן		Stop1	14	< -- @rows(Data1)					
8			Pic1	.....						
9										
10										
11	גלי									
12										
13	עופר									
14										
15	מרים									
16										
17										
18										
19										
20	Data1									

נניח שהרשימה כתובה על דף נייר ומבצעים את העבודה בעזרת עפרון. כשנתרגם את הדרישה לסדר פעולות נקבל:

1.1 גש אל Data1
1.2 גש אל השורה הראשונה
2.1 אם הסבב האחרון בוצע - הסתעף אל 3.1
2.2 אם הסבב האחרון לא בוצע -
2.2.1 כתוב בה את רצף התוים: ---
2.2.2 נוע לשורה הבאה
2.2.3 הסתעף אל 2.1
3.1 /rlrData1 ~
3.2 {home}{goto}Data1 ~
3.3 {quit}

בבוקר בבוקר מגיע הנהג למגרש החניה בקצה העיר, נכנס לאוטובוס שלו, מתניע ויוצא לתחנה הראשונה בסיבוב שלו. תחנה אחרי תחנה הוא עובר במסלול סיבובי שמחזיר אותו אחרי הסבב הראשון לתחנה הראשונה. בקופתו הצטבר סכום כסף ולפני

שימשיך בסבב השני הוא מוסר את הסכום לאחראי. וחוזר חלילה פעם אחרי פעם.

עד מתי? כמה פעמים יעשה הנהג סבב כזה לפני שיפסיק ויצא להפסקת הצהרים? אפשר לקבוע שיפסיק אחרי הסבב החמישי. ומדוע דווקא חמש פעמים? מפני שסבב ממוצע נמשך שעה ומשמרת הבוקר יוצאת מהתחנה הראשונה בשבע בבוקר ומסיימת בשעה שתים עשרה בצהרים - חמש שעות = חמש פעמים לבצע את הסבב.

ובכן, המרכיב הראשון של הלולאה הסידרתית (ראשון מבחינת סדר הפעולות) הוא ההגעה למקום וההתייצבות בנקודת ההתחלה. כך הוא אצל נהג האוטובוס וכך גם במחשב. לפיכך בקטע זה של הלולאה נצטרך להוליך את הסמן באחת הטכניקות שלמדנו בפרק א ולהציבו בנקודת ההתחלה.

עדין לא בוצע כל סבב ואנחנו נאמר שבשלב זה בוצע הסבב אפס פעמים.

המרכיב השני של הלולאה היא הפעילות שמניבה את התפוקה המבוקשת. בדוגמה של האוטובוס הדבר פשוט - נוסעים עולים ויורדים במהלך הסבב והוא מניב הכנסה שנמסרת לאחראי בסיום כל פעם. הדוגמה בתרגיל 5.01 מורכבת מעט יותר מכיון שהתפוקה בעת הסבב מותנית במצב בשטח באותו הרגע ולכן יש לנו שם לולאת התניה סידרתית.

האחראי לתחנה נמצא כבר במקום כשהנהג מגיע לקראת הסיבוב הראשון ורשום אצלו כי הנהג יחזקאל ביצע הבוקר את הסבב אפס פעמים. ברגע שיצא יחזקאל לדרך משנה האחראי את הרישום מאפס לאחד. משסיים יחזקאל את הסיבוב הראשון ומסר את הקופה רשם האחראי שתיים במקום אחד ויחזקאל יוצא לסיבוב השני. במילים אחרות האחראי ממלא פונקציה של מונה פעימות.

מונה הפעימות קופץ צעד אחד קדימה (ומכאן "לולאה סידרתית") לפני שיחזקאל יוצא לסיבוב הבא. אחרי כל קפיצה כזאת בודק האחראי אם מונה הפעימות מראה מספר שהוא יותר גדול מחמש ואם כך הוא הדבר אזי יחזקאל יכל לצאת להפסקת הצהרים.

היציאה הזאת מהלולאה להפסקת הצהרים היא המרכיב השלישי של נוהל הלולאה הסידרתית.

הקם בגיליונך את האינטרה מקרו מתרגיל 5.04.1 הפעל אותו בשלבים ולא בשלבים. ראה כיצד משתנה מונה הפעימות (Counter1).

בצע תרגיל מתקדם כדלקמן. בתרגיל 5.04.1 מתבצעת הולכת סמן בפועל כפי שלמדנו בפרק א. אבל בפרק א למדנו גם הולכת

סמן לוגי וראינו שהיא יותר מהירה מבחינת הביצוע בגיליון ומאפשרת ביצוע מגוון רחב יותר של מטלות בחיבור מקרו קטן יותר.

כתרגיל מתקדם - כתוב מחדש את האינטרה מקרו הנוכחי - השתמש בנוהלי הולכת סמן לוגי במקום בהולכת סמן בפועל. הפתרון מוצג כאן בהמשך בתרגיל 5.04.2.

תרגיל 5.04.2
מרכיבי הלולאה הסידרתית
הולכת סמן לוגי

	A	B	C	D	E	F	G	H	I	J
1			\z	{let Counter1,0} ~ {home}{goto}Data1 ~ {u 2}						
2			1	{let Counter1,Counter1+1} ~ {if Counter1>Stop1}{branch 2}						
3				{if @index(Data1,0,Counter1-1)<>""}{branch 1}						
4	דוד			{put Data1,0,Counter1-1,Pic1} ~ {branch 1}						
5			2	/rlrData1 ~ {quit}						
6	חנה		Counter1	0						
7	דן		Stop1	14	< -- @rows(Data1)					
8			Pic1	.....						
9										
10										
11	גלי									
12										
13	עופר									
14										
15	מרים									
16										
17										
18										
19										
20	Data1									

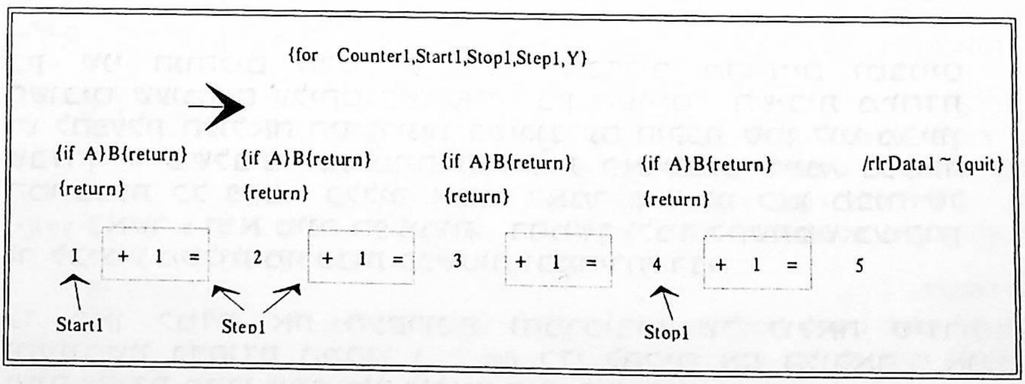
בין שני הנוהלים הללו יש מספר הבדלים מהותיים ומבניים חשובים שעומדים עליהם כשמשווים בין השניים. חשיבות מיוחדת יש להפעלת הלולאה הסידרתית בשילוב עם הולכת סמן לוגי מכיוון שבדרך זו משלבים את מונה הפעימות כארגומנט דינמי בפעולה המתבצעת כל פעם. כללית אפשר לאמר שיש לנו כאן מבנה של $y=f(x)$ כאשר x הוא מונה הפעימות. בהמשך נלמד להשתמש בעיקרון זה לביצוע מטלות מורכבות בפשטות ובמהירות רבה.

עד כאן למדנו את העקרונות והמרכיבים של לולאה סידרתית והשתמשנו בפקודות המקרו {let ... } כדי להקים את הלולאה. אולם ישנה פקודת מקרו שמפעילה לולאה סידרתית בצורה אוטומטית.

תרגיל 5.04.3
לולאה סידרתית עם {for ... }

	A	B	C	D	E	F	G	H	I	J
1			\z {home}{for Counter1,Start1,Stop1,Step1,1}/rlrData1 ~ {quit}							
2			1 {if @index(Data1,0,Counter1-1)<>""}{return}							
3			{put Data1,0,Counter1-1,Pic1} ~ {return}							
4	דוד		Counter1							
5			Start1	1						
6	חנה		Stop1	14	< -- @rows(Data1)					
7	דן		Step1	1						
8			Pic1							
9										
10										
11	גלי									
12										
13	עופר									
14										
15	מרים									
16										
17										
18										
19										
20	Data1									

המבנה והמטלות של תרגיל 5.04.3 מוכרים כבר משני התרגילים הקודמים. מהתרגילים הללו רואים בעליל שכל מרכיבי הלולאה הם למעשה ארגומנטים. ואמנם ישנה פקודת מקרו שמקימה ומפעילה לולאה סידרתית באופן מעשי כאשר כל המרכיבים מופיעים כארגומנטים. בפקודה זו מלת ההפעלה for מכניסה את המחשב לנוהל הלולאה ואחר כך יש חמישה ארגומנטים.



{for Counter,Start,Stop,Step,Range}

מונה פעימות — Counter

ארגומנט זה חייב להיות שם של תחום והוא מעין מכשיר מכני או אלקטרוני שסופר התרחשות סידרתית. דהיינו, כל פעם שמופעל משפט המקרו ששוכן בכתובת Range (האוטובוס מתחיל סבב) מוסיף המונה את המספר אחד למספר שכבר רשום בו. כלומר עם התחלת הלולאה (על ידי הפעלת הפקודה {for ... }) מתחיל המונה לרוץ מאפס. כתוצאה מכך מקבלים סידרת מספרים

0 , 1 , 2 , . . , n

וכבר ראינו למעלה כיצד אפשר להריץ סמן לוגי כפונקציה של מונה הפעימות.

את מונה הפעימות הזה אפשר להריץ במגוון רחב של צורות:

התחלה — Start

אפשר לאפס את המונה כך שיתחיל מאחד או מ 7 או מ 58 או מכל מספר אחר שרוצים. ארגומנט זה חשוב אם למשל רוצים להתחיל להפעיל את משפט המקרו מהשם ה 7 ברשימה. אין חובה להקים שם תחום לארגומנט הזה ואפשר לכתוב את המספר שרוצים כארגומנט השני. אולם אז מאבדים את יתרון המודולריות שמאפשר, בדוגמה זו, להתחיל כל פעם שמריצים את הלולאה משם אחר. לכן מומלץ תמיד להקים תחום גם לארגומנט הזה.

מעצור — Stop

בעת ביצוע הלולאה משווה המחשב ללא הרף את המספר שכתוב בארגומנט הזה עם המספר הרץ במונה הפעימות. ברגע שהמספר במונה הפעימות גדול מהמספר במעצור יפסיק המחשב לבצע את הלולאה וימשיך לקרוא את פקודות המקרו שכתובות באותה שורה ומימין ל {for ... }. כלומר, במעצור כותבים את מספר הפעמים שרוצים לבצע את הלולאה. גם כאן אפשר לכתוב את המספר במקום הארגומנט וגם כאן רצוי מאוד להקים שם תחום עבורו והארגומנט יכול להיות כל מספר או משוואה שרוצים.

צעד — Step

למעלה ראינו שהמונה סופר בקפיצות של אחת. אבל אפשר גם לכוון אותו כך שיספור בקפיצות של 2 או חמש או כל מה שרוצים. בדרך כלל רוחב צעד = 1 אבל אם נרצה לבצע את המקרו על כל שם שני נכתוב 2 בארגומנט הרביעי הזה.

כתובת למקרו — Range

הכתובת בה כתוב משפט המקרו שיופעל כל פעם. משפט זה הוא משפט שירות ולכן יסתיים בפקודה {return}.

הקם בגיליונך את האינטרה מקרו שבתרגיל 5.04.3. הפעל את המקרו מספר פעמים באופן רגיל ובשלבים. בצע שוב עם ערכים שונים בארגומנטים. לפני כל הפעלה כתוב "ציור" אחר ב Pic1 כדי לראות את המתרחש.

התרגיל הבא מלמד כיצד להפעיל לולאות משולבות אחת בשניה. התרגיל מוצג במספר שלבים: נתונים, הגדרת דרישות ופתרון.

תרגיל 5.05.1
לולאות משולבות להדפסת מדבקות
נתונים

	A	B	C	D	E	F	G
1							
2							
3		רשימת שמות וכתובות					
4							
5	יישוב	רחוב	שם				
6	========	==========	==========				
7							
8	תל אביב	הרצל 12	בן דוד				
9	ירושלים	בורלא 12	דן שמואלי				
10	ירושלים	יפו 87	ארקין שלמה				
11	תל אביב	הכרמלי 11	גלילי				
12	חיפה	גמלה 7	אשל זיוה				
13	רחובות	ת.ד. 103	גולני חנה				
14	רחובות	אילת 15	שובל בני				
15	נתניה	הארזים 77	דן חניתה				
16	תל אביב	גניורד ב	אסתר				
17							
18							
19	Data1						
20							

	D	E	E	F	G	H	I	J
1								
2		בן דוד						
3		הרצל 12	< ---	מבנה מדבקה				
4		תל אביב						
5								
6								
7		Print1						
8								

יש רשימת שמות וכתובות בתחום שנקרא Data1. הרשימה ערוכה במבנה של רשומות ושדות. רוצים לשלוף את הרשומות הללו ולהדפיס אותן על מדבקות במבנה של מדבקה. העתק את הנתונים אל הגיליון וכתוב את החיבור שיניב את התפוקה הזאת.

תרגיל 5.05.2
לולאות משולבות להדפסת מדבקות
פתרון

	G	H	I	J	L	M	N	O	P	Q
1	\z {for Counter1,Start1,Stop1,Step1,1}{quit}									
2	1 {for Counter2,Start2,Stop2,Step2,2}									
3	/rlrPrint1 ~ {wait @now+@time(0,0,02)}{blank Print1} ~ {return}									
4	2 {put Print1,0,@mod(Counter2,3),@index(Data1,2-@mod(Counter2,3),Counter1)} ~									
5	{return}									
6	Counter1									
7	Start1	0								
8	Stop1	8	<---	@rows(Data1)-1						
9	Step1	1								
10	Counter2									
11	Start2	0								
12	Stop2	2	<---	@rows(Print1)-1						
13	Step2	1								
14										

כתחליף לפקודת ההדפסה מופיעה כאן פקודת המקרו {... wait}. בהגיעו לפקודה זו ימתין המחשב את הזמן הרשום בארגומנט ורק אח"כ ימשיך לבצע את המשך משפט המקרו.

נתאר את סדר הדברים שצריכים להתרחש כדי שהמטלה הזאת תתבצע.

1. מוליכים סמן אל הרשומה הראשונה ברשימת השמות
2. מציבים סמן על השם ברשומה הראשונה
3. מעתיקים את השם אל השורה הראשונה במדבקה
4. מוליכים את הסמן הלוגי אל העמודה הבאה
5. מעתיקים את הרחוב והמספר אל השורה השניה במדבקה
6. מוליכים את הסמן הלוגי אל העמודה הבאה ב Data1
7. מעתיקים את שם היישוב אל השורה השלישית במדבקה
8. מישרים את הטקסט במדבקה לימין
9. מדפיסים את המדבקה
10. שואלים אם זאת היתה הרשומה האחרונה, אם כן מפסיקים אחרת
11. מוליכים את הסמן הלוגי אל הרשומה הבאה

וכך הלאה

{wait @now+@time(0,0,02)}

המתן

מהרגע הנוכחי [@now] 0 שעות, 0 דקות, 2 שניות [@time(0,0,02)]

מהתאור הזה נראה בברור שיש לנו כאן שתי לולאות - אחת בתוך השניה. הלולאה הראשונה מוליכה סמן לוגי על פני הרשומות ברשימת השמות. כל פעם שמגיעים לרשומה חדשה מתבצעת מטלת ההדפסה וכך עד סוף הרשימה.

גם מטלת ההדפסה מתבצעת על ידי לולאה. תחילה יש להעתיק את שם הנמען מהעמודה הראשונה ברשימת השמות אל השורה הראשונה בתחום שממנו תודפס המדבקה. אח"כ צריך להעתיק את הרחוב והמספר אל השורה השניה במבנה המדבקה ולבסוף מעתיקים את שם היישוב אל שורת המדבקה האחרונה ומדפיסים. עכשיו חוזרים ללולאה הראשונה וממשיכים עד שגומרים לעבור על כל השמות.

שתי פקודות המקרו { ... for} בחיבור זה מבצעות למעשה רק מעקב אחרי מספר הפעמים שכל דבר התבצע ומעדכנות את מוני הפעימות. העבודה העיקרית - הכנת המדבקה להדפסה - מתבצעת על ידי פקודת המקרו { ... put} בפיסקה 2 בחיבור המקרו.

```
{put  Print1,0,@mod(Counter2,3),@index(Data1,2-@mod(Counter2,3),Counter1)}~
```

נבחן צעד צעד את מבנה הארגומנטים בפקודה זו.

put	מלת ההפעלה - שים
Print1,	איפוא לשים - בתחום שנקרא Print1
0,	בעמודה מספר 0
@mod(Counter2,3),	בשורה שמספרה מחושב על ידי ------ <

הסבר

לכל נמען יש שלוש שורות במדבקה כמספר העמודות עם נתונים שיש לו ברשימה. לפיכך פעולת ההעתקה צריכה להתבצע 3 פעמים לכל נמען. בתרגיל זה מתחילים שני המונים מאפס ולא מאחד. מכיוון שכך נוסחת המעצור היא מספר השורות במדבקה פחות אחד. כתוצאה מכך יכול המונה השני לקבל רק את הערכים

0 1 2 3

כשהמונה השני מקבל את הערך 3 הלולאה השניה נפסקת וחוזרים לבצע את הלולאה הראשונה. לכן מטלת הלולאה השניה - העתקה - תתבצע רק כשלמונה ערכים 2, 1, 0.

כפי שראינו בפרק א הפונקציה @mod(x,y) מפיקה את השארית מביצוע X חלקי Y. ולכן ...

כאשר המונה השני הוא 0 מקבלים שארית = 0 מ 0/3 וזה יהיה מספר השורה במדבקה אליה יועתק השם. כשהמונה הוא 1 השארית של 1/3 היא 1 וזה יהיה עתה מספר שורת המדבקה שתקבל את הרחוב. וכשהמונה יהיה 2 יקלט היישוב בשורה האחרונה במדבקה.

@index(	מאיפה לקחת את הנתונים - נפעיל את הפונקציה --->
Data1,	בתחום שנקרא Data1 המכיל את רשומות הנמענים
2-@mod(Counter2,3),	נתוני הנמען בבסיס הנתונים כתובים בעברית מימין לשמאל. לכן השורה הראשונה במדבקה צריכה לקבל את הנתון מהעמודה האחרונה ברשימת השמות. כפי שראינו למעלה נותנת פונקצית השארית של המונה השני את הספירה בסדר עולה החל מאפס. לפיכך צריך להפחית את תפוקת משוואת השארית ממספר העמודות. כאשר עובדים עם מטריצות נתונים המסודרות משמאל לימין משתמשים במשוואת השארית בלבד.
Counter1)}	הארגומנט הראשון בפונקצית האינדקס הוא מספר העמודה ממנו ישלף הנתון והוא מתקבל מפונקצית השארית של המונה השני כפי שראינו זה עתה. הארגומנט השני מציין את מספר השורה. כאן הוא פשוט הערך הנוכחי של המונה הראשון. מונה זה מתקדם פעם אחת עם כל פעם שמודפסת מדבקה. ומשום שהלולאה הראשונה מתחילה גם היא מאפס אזי המדבקה הראשונה תקבל את הנתונים של הנמען הראשון ברשימה וכך הלאה.

בסעיף 5.2.4 בעמוד 104 הצענו לתלמיד תרגיל חשיבה ופיתוח. עתה יש בידינו כל הכלים לפתרון.

נתון גיליון אלקטרוני ובו לוח עם נתונים למשל, כל שורה היא נקודת יישוב בארץ ובעמודות יש פרטים כמו שם היישוב, מספר התושבים, מספר בתי הספר או כל מה שתרצו.

הלוח ממוין מראש לפי סדר הא"ב של השמות.

רוצים להוסיף עוד שורה בשביל יישוב חדש.

רבים פשוט מוסיפים שורה בסוף וממיינים מחדש. אלא שמיון של לוחות גדולים לוקח זמן רב ויכולים להווצר שיבושים במשוואות ובהגדרות התחומים.

נוהל חלופי הוא "לרוץ" על פני השמות, לגלות את המקום המדויק בו צריך השם החדש להופיע ולהכניס שורה ע"י ~/wir.

כתוב את חיבור האינטרה מקרו לביצוע המטלה הזאת כאשר בסיס הנתונים הנתון הוא בן עמודה אחת ומספר שורות

תרגיל 5.06
הוספת רשומה לפי הא"ב ללא מיון

	A	B	C	D	E	F	G	H	I
1	\z {getlabel Msg1,Name1} ~ {for Counter1,Start1,Stop1,Step1,1}{branch 2}								
2	1 {recalc Adr1} ~								
3	{if @index(Data1,0,Counter1)<=Name1}{return}								
4	/wir{Adr1} ~ {3}{branch 4}								
5	2 /rncData1 ~ {d} ~ {branch 4}								
6	3 {if Counter1>0}{return}								
7	{goto}Data1 ~ {u}/rncData1 ~ {bs}.{d}{end}{d} ~ {return}								
8	4 {put Data1,0,Counter1,Name1} ~ /rlrData1 ~ {quit}								
9	Name1								
10	Counter1	0							
11	Start1	0							
12	Stop1	8 < ---- @rows(Data1)-1							
13	Step1	1							
14	Msg1	Alt + Tab...שם פרטי לעברית -»							
15	Adr1	d17{return}	< -- +"d"&@string(@cell("row",Data1)+Counter1,0)&"{return}"						
16									
17				דוד					
18				חיים					
19				טובה					
20				ישראל					
21		Data1 --- >		כוכבית					
22				מרים					
23				נועם					
24				נעמה					
25				פנינה					
26									

כשיש רשימת שמות ממוינת לפי סדר הא"ב ורוצים להוסיף שם חדש בוחנים שורה שורה ומוצאים את השורה שהשם הכתוב בה יותר גדול (במשמעות של ב > א) מהשם החדש. בשורה זו מבצעים Worksheet Insert Row וכותבים בשורה החדשה את השם החדש.

אם השם החדש צריך להיות ראשון או אחרון ברשימה נפעל בכל מקרה בצורה שונה.

התקן את האינטרה מקרו הזה בגיליון והפעל אותו לקליטת שמות שונים. עבור עליו בעיון ומצא את כל מה שמתרחש כפונקציה של הערך הנוכחי במונה הפעימות.

מונה הפעימות מספק מידע רב לגבי התרחשויות שונות במהלך ביצוע הלולאה הסידרתית. השווה את מה שגילית עם הרשימה הבאה.

רשימה זו מתארת את הפונקציות של מונה הפעימות לפי סדר הופעתן במקרו.

1. ארגומנט ב { ... for} לשם השוואה עם המעצור
2. משתנה במשוואה שכתובה ב Adr1. בצורה זו מקבלים את כתובת התא הנוכחי לצורך ביצוע פקודות תפריט כמו /wir מבלי להציב שם בפועל את הסמן.
3. ארגומנט בפונקצית index לשליפת תוכן התא לצורך השוואתו עם השם החדש.
4. מבחן בהתניה הבודקת אם השם החדש צריך להיות ראשון ברשימה
5. ארגומנט במקרו { ... put} כדי לדעת איפה לכתוב את השם החדש

כיצד יודע המחשב שהשם החדש צריך להיות האחרון ברשימה?

אם הלולאה הסתיימה ולא נמצא ברשימה שם יותר גדול מהשם החדש אזי השם החדש הוא הגדול ביותר וצריך להיות אחרון. משפט 2 מבצע את הנוהל המקרה כזה וההפניה אליו על ידי {branch 2} מתבצעת אחרי היציאה מהלולאה.

בתרגיל 5.06 חייבים לבצע recalc על Adr1 לפני שממשיכים כדי שהכתובת תתעדכן אחרי קפיצת מונה הפעימות.

משתמשים רבים רוצים להדפיס דורות שונים מנתונים בגיליון אחד. לאחר עיבוד הנתונים בגיליון מתקבלים דוחות שונים. כל דוח במבנה שונה, כל דוח במקום אחר בגיליון ולכל דוח שם תחום משלו.

תרגיל 5.07 מדגים חיבור אינטרה מקרו לביצוע המטלה הזאת. חיבור זה משלב מספר נוהלים, כל אחד מהם נלמד עד כה בנפרד.

* לולאה סידרתית על ידי { ... for}
* ארגומנט דינמי מוחדר לתוך משפט מקרו כדי להזין כל פעם את שם התחום הבא להדפסה
* שליפת שם לוח להדפסה כפונקציה של מונה הפעימות
* אם נפסקת ההדפסה עקב טעות מדפסת (נגמר הנייר למשל) עצירת המקרו ואחרי תיקון הטעות המשכת ההדפסה - אוטומטית - מהמקום שהסתיימה

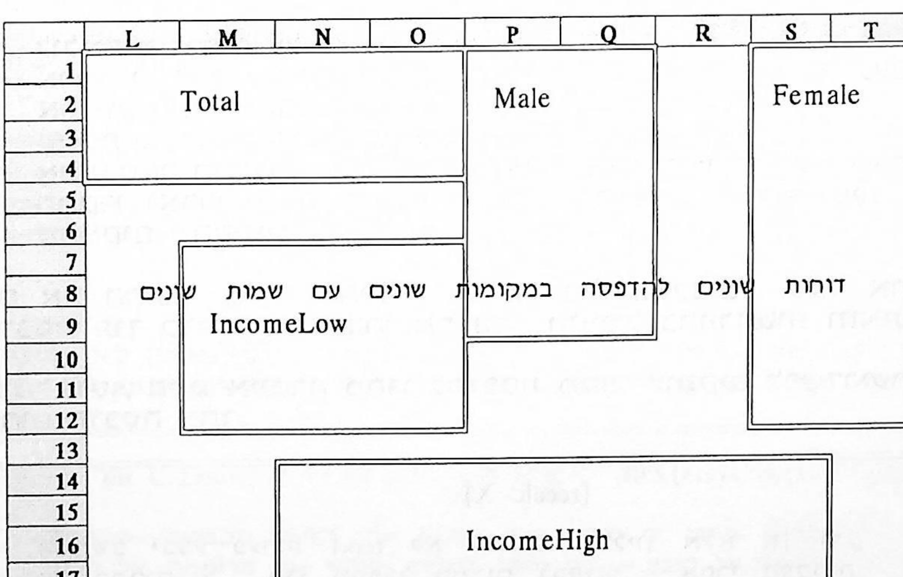
הקם את החיבור בגיליון והפעילו באופן רגיל ובשלבים. סגור את המדפסת תוך כדי הדפסה ובחן את נוהל הטיפול בהתרחשות הזאת.

כתרגיל נוסף, הקם אינטרה מקרו להדפסת מספר עותקים לפי דרישה מתחום הדפסה אחד.

> {recalc X}~
>
> המחשב יבצע פעולת {calc} לא על כל הגיליון אלא אך ורק על התחום X. כדי שתפעל חייבים לכתוב ~ אחרי הפקודה.

תרגיל 5.07
הדפסת דוחות מתחומים שונים בגיליון אחד

	A	B	C – J
1		\z {let Start1,1}~	
2		1 {for Counter1,Start1,Stop1,Step1,2}~{quit}	
3		2 {recalc Print1}~{onerror 3,Msg1}/ppr{Print1}~agpq{return}	
4		3 {let Start1,Counter1}{let Msg1,Msg1&Msg2}~{indicate Msg1}{Beep1}{?}{indicate}{branch 1}	
5	Beep1	{beep 1}	{beep 1}{beep 1}{beep 2}{beep 2}{return}
6	Print1	ERR	< --- @vlookup(Counter1,Data1,1)&"{return}"
7	Counter1		
8	Start1	1	
9	Stop1	5	< --- @rows(Data1)
10	Step1	1	
11	Msg1		
12	Msg2	... ENTER בדוק מדפסת ואח"כ הקש	
13			
14			1 Total
15			2 Male
16	Data1 →		3 Female
17			4 IncomeLow
18			5 IncomeHigh
19			
20			



פרק ו: שו״ב - שליטה ובקרה

6.1 הבעיות בשטח

הגיליון האלקטרוני הוא כלי עבודה המגביל את התפוקה אך ורק ביכולתו של המבצע. והדברים אמורים במגוון הדברים השונים שניתן לבצע, בסוגי הביצועים וברמתם המקצועית. מטבע הדברים יש כאן תנאי חממה להיווצרות כמויות אדירות של חומר ברבדים שונים. ברובד התוך גיליוני הולכים הקבצים וטופחים בגודל ובמורכבות עם הצטברות חומר ועם התפתחות התיחכום בדרישות העיבודים. ברובד הבין גיליוני הולך וטופח מספר הקבצים לכמויות גדולות ביותר.

התייחסות אל מקרו בגיליון האלקטרוני כאל תורת אוטומציה מאפשרת לנו לטפל בבעיות אלה. עד כאן למדנו את יסודות תורת המקרו. ויסודות אלה הם התשתית לכל מה שנלמד מכאן ואילך.

עד כה למדנו איך לשפר ביצועים בגיליון וכיצד לבצע מספר רב ומגוון של מטלות בגיליון שלא היו ניתנות לביצוע בלי מקרו. למדנו גם נוהלים לשליטה ובקרה תוך גיליונית כמו אירגון הגיליון במבנה אינטרה מקרו והפעלת משפטים אינטראקטיביים. השליטה הבין גיליונית (על קבצים ובתוך קבצים בדיסק) באמצעות הגיליון שעל המסך חובקת עולם ומלואו ושם באמת מתגלה כוחו העצום של המקרו בגיליון (ראה סיפרי "מקרו למתקדמים בלוטוס" בהוצאת הוד-עמי) והיא דורשת ידע בשילוב נוהלי ASCII בגיליון ועל הדיסק. לפיכך נעסוק בפרק זה בקטע אחד בלבד מתוך המכלול הזה. לא נעסוק בנוהלי ASCII אלא אך ורק בקבצי worksheet, אירגון הקבצים בדיסק והשליטה בקבצים הללו.

6.2 אירגון הדיסק לשו״ב בקבצי worksheet

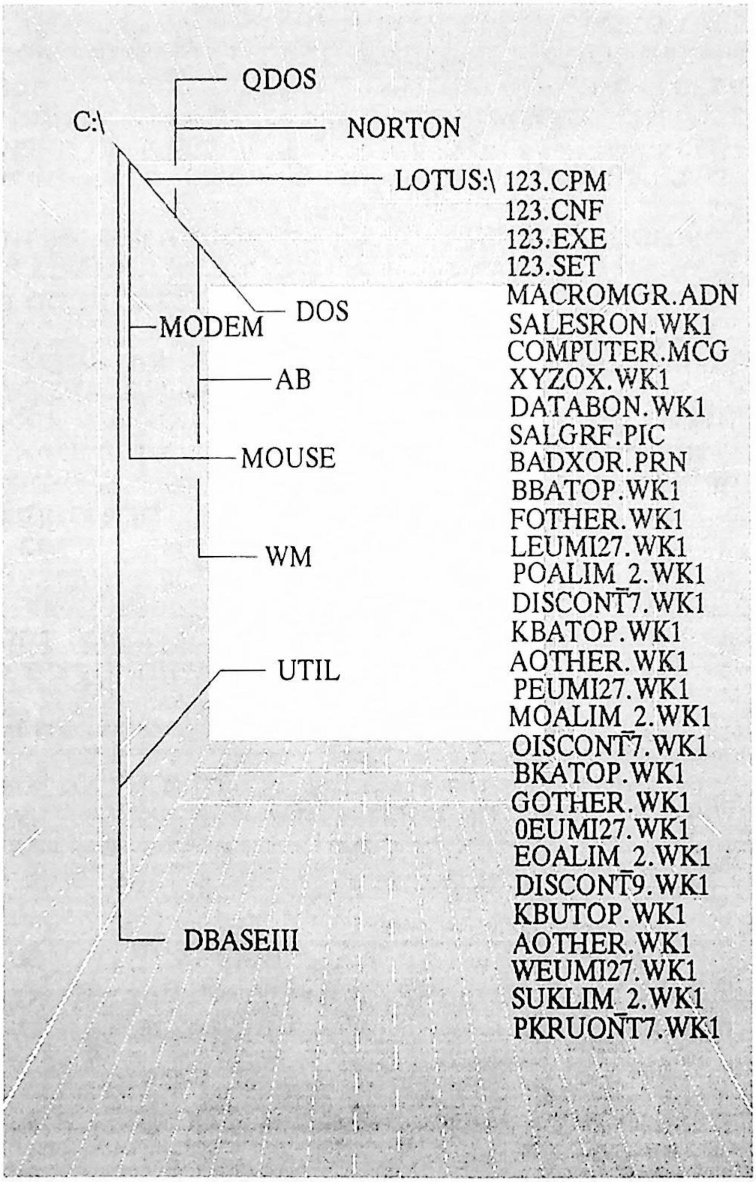
איור 6.01.1 מציג תמונה לא מלבבת ביותר של דיסק קשיח. תמונה זו אופינית למשתמש פעיל שעובד הרבה עם המחשב אבל יש לו בעיות שליטה בקבצים שלו. נקח לדוגמה את מחיצת הלוטוס בדיסק הזה. בכפיפה אחת שוכנים שם קבצים מסוגים שונים - קבצי תוכנה, גרפיקה, גיליונות עבודה ומה לא.

נתמקד לרגע רק בקבוצת קבצי wk1 שבמחיצה זו. יש כאן 23 קבצים כאלו, מספר לא רב למשתמש פעיל. משתמשים לא מעטים מגיעים גם ל 100 קבצים ויותר במחיצה אחת כשהם מעורבים עם קבצי

```
{system "dir C:\Lotus\*.* > C:\Y.prn"}{goto}Data1~/fitX{esc}C:\y.prn~{quit}
```

משפט מקרו זה יצא לDOS, יעתיק את רשימת הקבצים שבמחיצת הלוטוס אל קובץ C:\Y.prn, יחזור לגיליון וישלוף את הרשימה אל המסך.

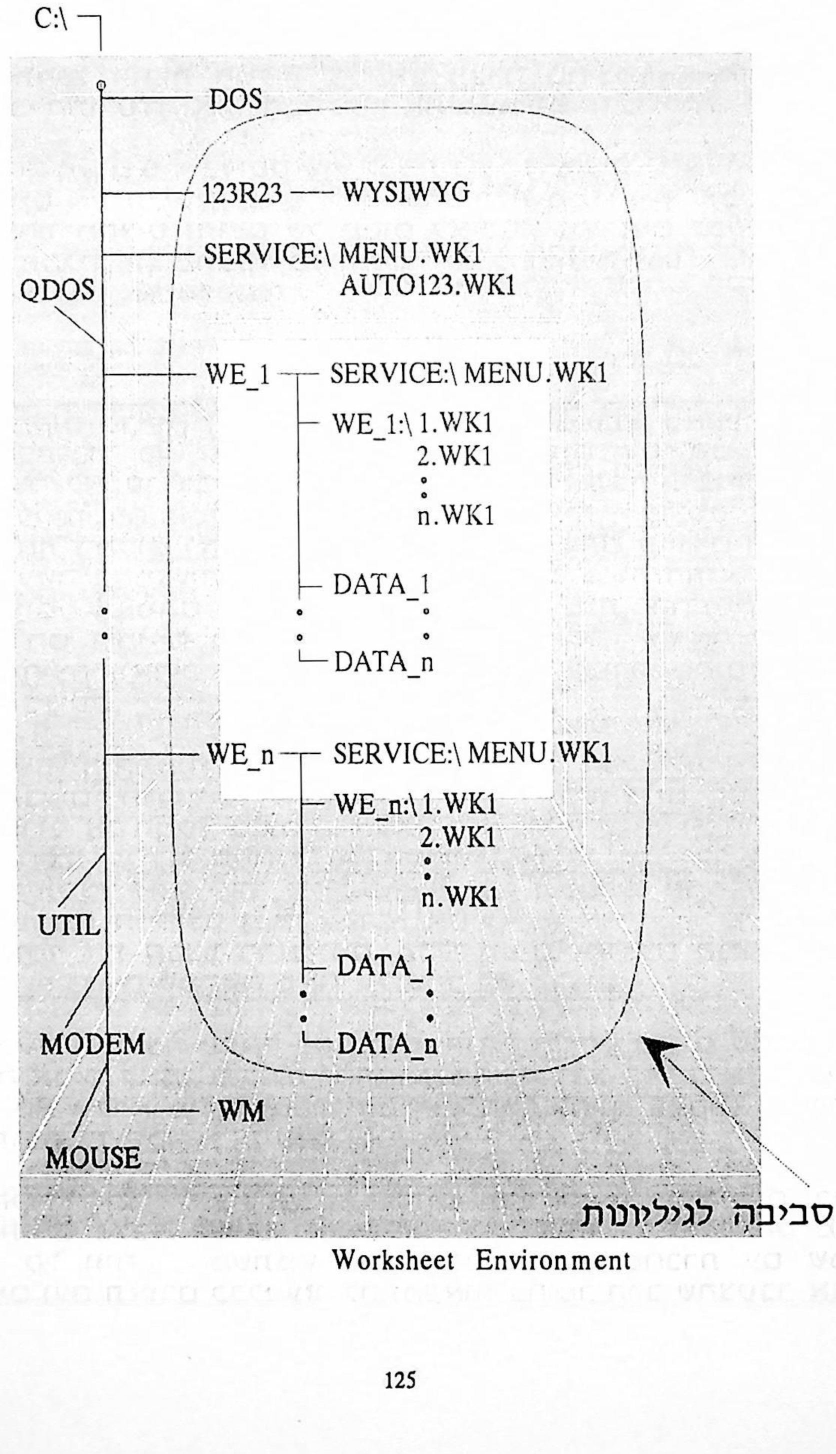
איור 6.01.1
מרחב הדיסק
לא מאורגן



איור 6.01.2

מרחב הדיסק

מאורגן



Worksheet Environment

התוכנה של הגיליון האלקטרוני, קבצי תכנה לתוספים, קבצי גראפיקה, קבצי טקסט ואחרים. מה הביא קבצי הגיליון למחיצה זו?

6.2.1 מקומות וכתובות בדיסק

בעת התקנת תוכנת הגיליון במחשב נוצרת מחיצה Directory המכילה את קבצי התוכנה הזאת. הבה נבחן את משמעות הדבר הזה.

מחיצה זה מקום פיזי בדיסק שיש לו כתובת חד ערכית. האיורים למעלה זה מציגים את הדיסק כאילו היה מרחב תלת ממדי אין סופי. במרחב הזה אנחנו מקצים נתחים של מקום לאיכסון וקוראים לכל נתח כזה בשם. תהליך זה מתבצע כשנותנים את פקודת ה DOS - md. למשל כשניתן ב DOS את הפקודה

```
md c:\123r23
```

יוקצה מקום בדיסק למחיצה זו. כלומר המחשב יבחר באופן אוטומטי במקום/כתובת כל שהם בדיסק ויקשר אותם עם שם המחיצה. אי אפשר לאכסן דברים אם אין מקום עם כתובת. בגיליון אנחנו רואים פיזית את המקום ואת הכתובת בו-זמנית. בדיסק אנחנו לא רואים את המקום ואת הכתובת - רק יודעים שהם קיימים וכי הקשר איתם נעשה באמצעות שם המחיצה. לפיכך כל שמות התואר, הפעלים והביטויים למיניהם שישמשו אותנו כאן בעת ההתייחסות אל הדיסק הם דימויים בלבד. הדבר הממשי היחידי שנראה לעין הוא שם המחיצה (ואח"כ גם שם הקובץ) שמייצג עבורנו מקום וכתובתו אי שם בדיסק.

נאמר שבעת התקנת התוכנה נוצרה המחיצה הנ"ל ולתוכה הועתקו כל הקבצים הדרושים להפעלת הגיליון. כשמעלים את הגיליון האלקטרוני אל המסך בפעם הראשונה הוא מתייחס אל המחיצה הזאת כאל מחיצת ברירת המחדל של הגיליון. פרושו של דבר הוא שכל אימת שנבצע File Save או File Retrieve תתבצע הפעולה באופן אוטומטי עם המחיצה c:\123r23 אלא אם נאמר אחרת. לכן, משתמש חדש שומר את קבציו הראשונים, מבלי משים, יחד עם קבצי התוכנה ונוצרת עירבוביה שמקשה מאוד על שו"ב בין קבצים.

ממש כשם שמחיצה מזוהה עם מקום/כתובת בדיסק כך גם שם הקובץ. למעשה אפשר לאמר שיכולה להתקיים מחיצה ללא קבצים אך לא יכול להיות קובץ ללא מחיצה מכיון שהקובץ חייב להיות במקום כל שהוא עם כתובת בדיסק. וכאן נתקלים בקושי נוסף.

שם הקובץ יכל להיות בן 11 תווים לכל היותר ובמקרים רבים בן 8 תווים בלבד. מספר תווים זה קטן ואיננו מכיל מספיק מידע לזיהוי קל ונוח. המשתמש נאלץ לכן להחזיק מחברת עם שמות הקבצים ועם תאורים ככלי עזר להתמצאות בחומר הרב שהצטבר אצלו.

אין כל ערך לנתונים המאוכסנים אי שם בדיסק ולא ידוע "מה יש שם". למעשה יש להם ערך שלילי מכיוון שהם מחזיקים מקום ותורמים לעירבוביה. נתונים זהים מוחזקים במספר קבצים, בחלקם מעודכנים ובחלקם לא מעודכנים והשימוש בהם עשוי לגרום טעויות קריטיות. לא מעטים הם המשתמשים שמפעילים אחוז קטן מאוד מהקבצים שהקימו ואילו את כל היתר "אבדוק כשיהיה לי זמן".

כדי להכניס סדר בכל אלה, כדי להקל על החיים וכדי לאפשר פיתוחים חכמים בעתיד נבצע את הדברים הבאים.

1. בעזרת תורת המקרו שלמדנו עד כה נפתח קובץ קישור בינינו לבין הדיסק - תוכן עניינים במקום fr/.

2. נארגן את קבצי ה wk1 בדיסק במבנה של סביבות עבודה.

3. נקשר בין תוכן העניינים לבין סביבות העבודה שייצרנו.

6.2.2 תוכן עניינים במקום fr/

נדמיין לעצמנו ספר בלי תוכן עניינים. בספר כזה צריך לקרוא הכל מהתחלה כדי למצוא דבר מה. נדמיין גם ששמות הפרקים בספר כתובים בקוד בן שמונה ספרות ו/או אותיות והרי לנו המצב כפי שהוא מתואר באיור 6.01.1.

בפרק א סעיף 1.3.2 (עמוד 49) למדנו לבנות מקרו אינטראקטיבי והקמנו תוכן עניינים להולכת סמן בגיליון. נשתמש באותו העקרון להקמת תוכן עניינים לקבצים שלנו. לשם כך נניח שסביבת העבודה לגיליונות אלקטרוניים בדיסק שלנו נראית כמו במיפוי הזיכרון באיור 6.02.

בתרגיל 6.01 נלמד להקים חיבור אינטרה מקרו שנועד לשמש כתוכן עניינים לקבצים שהקמנו בגיליון האלקטרוני. לשם כך נייצר מספר קבצי wk1 וקובץ תוכן עניינים שיקשר ביניהם.

מקרו בשם: \0

ברגע שהגיליון מגיע למסך מחפש המחשב, באופן אוטומטי, מקרו בשם \0. אם יש מקרו בשם כזה המחשב מפעיל אותו מיד.

איור 6.02
מפת הדיסק לסביבות עבודה בגיליונות אלקטרוניים

Directories - מחיצות			Files - קבצים	
C:\	123r23			מחיצה לקבצי ההפעלה של תוכנת הגיליון בלבד
	Service:\		Auto123.wk1	תוכן עניינים לברירת מחדל
			Menu.wk1	תוכן עניינים לתיקשורת בין סביבות
	WE_1	Service:\	Menu.wk1	תוכן עניינים לתיקשורת בתוך הסביבה
		WE_1:\	1.wk1	גיליון לתפעול
			2.wk1	גיליון לתפעול
			3.wk1	גיליון לתפעול
		Data_1:\	1.wk1	גיליון לנתונים
			2.wk1	גיליון לנתונים
			3.wk1	גיליון לנתונים
			4.wk1	גיליון לנתונים
		Data_2:\	1.wk1	גיליון לנתונים
			2.wk1	גיליון לנתונים
	WE_2	Service:\	Menu.wk1	תוכן עניינים לתיקשורת בתוך הסביבה
		WE_2:\	1.wk1	גיליון לתפעול
			2.wk1	גיליון לתפעול
			3.wk1	גיליון לתפעול
		Data_1:\	1.wk1	גיליון לנתונים
			2.wk1	גיליון לנתונים
			3.wk1	גיליון לנתונים
			4.wk1	גיליון לנתונים
		Data_2:\	1.wk1	גיליון לנתונים
			2.wk1	גיליון לנתונים

WE_1: סביבת עבודה למטלה מספר 1

WE_2: סביבת עבודה למטלה מספר 2

Auto123.wk1

תוך כדי טעינת תוכנת הגיליון פונה המחשב אל מחיצת ברירת המחדל ומיד מחפש שם קובץ בשם Auto123.wk1. אם לא קיים קובץ כזה מקבלים על המסך גיליון אלקטרוני חדש. אם הקובץ קיים הוא נשלף באופן אוטומטי אל המסך.

תרגיל 6.01.1
הקמת קובץ תוכן עניינים
אירגון תשתית

1. פתח את המחשב והשאר ב (או עבור אל) מסך DOS בכונן C

2. כתוב

md c:\Service

ואחר כך הקש ENTER כדי לייצר מחיצה בשם c:\Service

3. כתוב

md c:\WE_1

ואחר כך הקש ENTER כדי לייצר מחיצה בשם c:\WE_1

4. הפעל את תוכנת הגיליון האלקטרוני שלך. על המסך יהיה גיליון חדש וריק לגמרי

5. כתוב את חיבור המקרו לפי הדוגמה שלהלן

	A	B	C	D	E	F	G	H	I	J
1				\m /frMenu~{quit}						
2										

6. החלף ידנית את מחיצת העבודה שלך ל c:\WE_1 על ידי ביצוע

/fdC:\WE_1\

7. כתוב את הסיפרה 1 בתא b2 ושמור את הקובץ תחת השם 1.wk1

8. כתוב את הסיפרה 2 בתא b2 ושמור את הקובץ תחת השם 2.wk1

9. כתוב את הסיפרה 3 בתא b2 ושמור את הקובץ תחת השם 3.wk1

10. מחק את הגיליון מהמסך על ידי ביצוע ידני של הפקודות

/wey

11. החלף ידנית את מחיצת העבודה שלך ל c:\Service על ידי ביצוע

/fdC:\Service\

12. צייר את מפת הדיסק לגיליונות אלקטרוניים שקיבלת

WE_1 = Worksheet Environment number 1

איור 6.03
מפת הדיסק שנוצרה בתרגיל 6.01

```
c:\ ──── ? ? ?        (Lotus כתוב כאן את שם מחיצת הגיליון שלך (למשל
   │
   ├──── Service
   │
   └──── WE_1:\   1.wkl      קובץ גיליון מספר 1
                  2.wkl      קובץ גיליון מספר 2
                  3.wkl      קובץ גיליון מספר 3
```

עכשיו הקם את האינטרה מקרו שבתרגיל 6.01.2.

תרגיל 6.01.2
הקמת קובץ תוכן ענייניים
חיבור מקרו שילדי

	A	B	C	D
1	תרגיל אירגון	תוכן העניינים		
2	סביבת עבודה	[ ]		
3		קובץ גיליון מספר 1	פעילות:	{bs}WE_1\1
4		קובץ גיליון מספר 2		{bs}WE_1\2
5		קובץ גיליון מספר 3		{bs}WE_1\3
6				
7	Data1	זמנית למסך DOS	יציאה:	{esc 6}/s{branch \z}
8		לגיליון חדש		{esc 6}/wey{quit}
9		מלוטוס		{esc 6}/qyy{quit}
10				

רוחב עמודה = 18 | רוחב עמודה = 34 | רוחב עמודה = 20

איזור הפעילות הגלויה: סה"כ = 72 = רוחב מסך שלם

	G	H	I	J	K	L	M	N	O
1	\z \0	{home}{?}/c{bs}{r 2}~xData1~{onerror \z}{if @cell("type",xData1)="b"}{branch \z}							
2		/fr							
3	xData1	[ ] <-- תא ריק							
4		~{branch \z}							
5	\a	/wtc{quit}							
6	\b	{home}{goto}Data1~/wtb{quit}							
7									

שים לב
המשפט הראשי מקבל שני שמות
האחד הוא \0 והשני הוא \z

איזור המקרו

תורת המקרו בלוטוס ותואמיו חלק א: יסודות

1. הקש על צמד המקשים Alt b
2. בצע שמירת הקובץ תחת השם Menu כדלקמן
/ f s M e n u
3. בצע שמירת הקובץ תחת השם Auto123 כדלקמן
/ f s A u t o 1 2 3
4. הקש על צמד המקשים Alt z
5. הצבע עם הסמן על "קובץ גיליון מספר 1" (תא b3) והקש ENTER
6. הקש על צמד המקשים Alt m
7. הצבע עם הסמן על "יציאה זמנית למסך DOS" והקש ENTER
8. בצע את פקודת הדוס
dir c:*.*
9. כתוב
e x i t
10. הקש ENTER
11. הקש על צמד המקשים Ctrl Break
12. הקש Esc
13. הקש על צמד המקשים Alt a
14. החלף את מחיצת ברירת המחדל של תוכנת הגיליון כך שתהיה
c : \ S e r v i c e

פעולה זו מתבצעת על ידי סידרת הפקודות
/Worksheet Global Default Directory
ובמהלכה ...
מקישים Esc כדי לבטל את ברירת המחדל הנוכחית ואח"כ מקישים c:\Service ו ENTER. לבסוף נותנים את הפקודות Update ו Quit.

15. הקש על צמד המקשים Alt b
16. הקש על צמד המקשים Alt z
17. הצבע עם הסמן על "יציאה מלוטוס" והקש ENTER
18. טען שוב את תוכנת הגיליון האלקטרוני שלך

אם כל הפעולות עד כאן בוצעו נכון יראה על המסך קובץ תוכן העניינים שהקמת כשהוא מוכן לפעולה. עתה, כדי לשלוף קובץ או כדי לבצע אופצית יציאה מצביעים עם הסמן על האופציה המבוקשת ומקישים ENTER.

מכיוון שקובץ זה תמיד מופיע כשהוא נשלט על ידי משפט המקרו (מדוע?) צריך לבטל את השליטה הזאת לפני שרוצים להכניס בו שינויים (כיצד?).

אחרי שיצאנו משליטת המקרו נפעיל Alt a כדי לפתוח את כל מה שסגרנו לפני ההפעלה כדי שאפשר יהיה לנוע חופשי ולכתוב בכל מקום.

אחרי הכנסת השינויים נפעיל Alt b כדי לסגור את הקובץ - לבצע את כל הפעולות הדרושות לפני הפעלה.

לבסוף נשמור את הקובץ פעמיים. פעם אחת תחת השם Auto123 ופעם אחת תחת השם Menu.

ועכשיו נפעיל Alt z ותוכן העניינים שלנו שוב נשלט על ידי מקרו ומוכן להפעלה על ידי הצבעה והקשת ENTER.

קיבלנו תשתית לקישור בינינו לבין קבצי הגיליון שלנו המאפשרת מעבר מקובץ לקובץ על ידי השילוב של קובץ עניינים מרכזי עם מבנה סביבתי עם מקרו לשליפת קובץ תוכן העניינים בכל קובץ גיליון.

במהלך עבודה שוטפת יש צורך באופציות יציאה שונות. בתרגיל הנוכחי ראינו שלוש אופציות. היציאה ל DOS מתבצעת על ידי הפקודה /s שמוציאה אותנו למסך המערכת וחוזרים לגיליון ע"י כתיבת Exit והקשת ENTER. כמו כן ראינו יציאה אל גיליון חדש ויציאה מתוכנת הגיליון. הפקודה {system ... } נותנת אופצית יציאה נוספת.

נאמר שיש לנו מעבד תמלילים WordMill וכי כדי לטעון אותו

קובץ בשם Auto123.wk1 עם מקרו \0

בעת טעינת תוכנת הגיליון האלקטרוני ולפני שמגיע אל המסך גיליון חדש, סורקת התוכנה את מחיצת ברירת המחדל שלה. אם במחיצה זו נמצא קובץ בשם Auto123.wk1 הוא נשלף אוטומטית אל המסך, אחרת מקבלים גיליון נקי. כאשר קובץ זה מכיל אינטרה מקרו לתוכן עניינים המופעל מיד בהגיעו אל המסך ע"י מקרו אפס (\0) מקבלים שרשרת אוטומציה לשליטה בכל סביבת העבודה בגיליונות.

אל המסך מקישים WM מכל מסך DOS שהוא. אזי אפשר להוסיף לתוכן העניינים בראש פרק "יציאה" את טעינת מעבד התמלילים

7	יציאה: זמנית למסך DOS	{esc 6}/s{branch \z}
8	לגיליון חדש	{esc 6}/wey{quit}
9	למעבד תמלילים	{esc 6}{system "WM"}{branch \z}
10	מלוטוס	{esc 6}/qyy{quit}

כאשר נבקש "יציאה למעבד תמלילים" יעלם מסך הגיליון ויופיע המסך של מעבד התמלילים ונוכל לעשות בו כל מה שנרצה. כאשר נצא ממעבד התמלילים נחזור ונקבל את מסך תוכן העניינים בגיליון באופן אוטומטי ונוכל להמשיך בפעילות השוטפת בגיליונות.

ולסיום נעבור שוב על סדר הפעולות של "המקרו השולף".

```
\z \0 {home}{?}/c{bs}{r  2}~xData1~{onerror  \z}{if  @cell("type",xData1)="b"}{branch  \z}
      /fr
      [      ] < --- xData1 תא ריק ושמו
      ~{branch \z}
```

משפט מקרו זו מופעל אוטומטית ברגע שהקובץ נטען למסך מכיוון ששמו \0. כמו כן אפשר להפעילו ידנית ע"י Alt z.

כשמציבים את הסמן על תאור של קובץ ומקישים ENTER מעתיק המחשב את השם "המחשבי" של הקובץ אל תא ריק בתוך משפט המקרו ששמו xData1. אם לאחר מכן תיווצר טעות מהסוג שמתגלה ע"י {onerror ... } תתרחש התפצלות חזרה אל תחילת המשפט. כלומר ניתנת למפעיל הזדמנות נוספת. התפצלות כזאת תתרחש גם אם התא xData1 הוא מסוג "b" (תא ריק = b = blank). למקרים שהוקש ENTER על תא ריק.

* * *

> ספר זה נועד ללימוד עצמי של יסודות המקרו ולפיכך עסקנו כאן בנושא שליטה ובקרה על קצה המזלג בלבד. ניצול מלא של יכולות הגיליון לשליטה בדיסק דורש ידע נוסף ומתקדם. ומי שרוצים להמשיך וללמוד עומד לרשותם ספרי
>
> מקרו למתקדמים בלוטוס
>
> בהוצאת הוד-עמי

תורת המקרו בלוטוס ותואמיו חלק א: יסודות

אל רן אברהמי
ת.ד. 4305
ירושלים 91042
טלפון 02-631768

אבקשך לעדכן אותי בפירטי ימי עיון וקורסים המועברים על ידך.

שם משפחה ________________ שם פרטי ________________

כתובת בעבודה

שם מקום העבודה ________________________________

אגף/מחלקה ________________________________

תפקיד ________________________________

רחוב ומספר ________________ יישוב ____________ מיקוד ________

טלפון מספר ________________ איזור ______ פקס ____________

כתובת בבית

רחוב ומספר ________________ יישוב ____________ מיקוד ________

טלפון מספר ________________ איזור ______ פקס ____________

רן אברהמי

תורת המקרו בלוטוס ותואמיו

חלק א

יסודות

ספר ללימוד עצמי

ספר זה נועד ללימוד עצמי של יסודות המקרו בגיליונות אלקטרוניים תואמי לוטוס 1-2-3. אין צורך בידע מוקדם במקרו ומספיקה רמת ידע סבירה בהפעלת הגיליון האלקטרוני.

החומר הנלמד בספר מתאים לכל גירסאות לוטוס 1-2-3 החל בגירסה 2.2 ומעלה. הוא גם מתאים, ברובו המכריע, לגירסת לוטוס 2.01

החומר מתאים לכל תוכנה התואמת לוטוס 1-2-3

הספר מכיל הסברים, תרגילים ואיורים המוליכים את התלמיד בהדרגה מהקל אל הכבד ומהפשוט אל המורכב בצורה קלה ונוחה ללימוד עצמי בבית.

בנוסף לפקודות המקרו לומדים ...

** לכתוב משפטי מקרו
** לכתוב חיבור מקרו
** אירגון הגיליון למקרו
** שילוב נוסחאות במקרו
** שילוב פקודות תפריט במקרו
** תיכנון לוחות ושילובם במקרו
** הגדרת צרכים ודרישות לקראת תיכנון מקרו
** הולכת סמן אוטומטית בגיליון
** ניהול בסיס נתונים - Data Query
** הקמת תפריט פקודות "עשה זאת בעצמך"
** מקרו ל ASCII
** לולאת התניה
** לולאה סידרתית
** לולאה בתוך לולאה
** אירגון הדיסק לעבודה עם קבצי wk1 רבים
** תוכן עניינים לניהול קבצי wk1 בדיסק ושליפתם
** שויב -- שליטה ובקרה

המחבר, רן אברהמי LAC ,MBA, מנתח ומלמד קורסים וימי עיון למשתמשי גיליונות אלקטרוניים ומפתח יישומי לוטוס במערכות מורכבות. רן הוא גם מחברו של הספר "מקרו למתקדמים בלוטוס" בהוצאת הוד-עמי.

LAC - הכרה מקצועית על ידי חברת לוטוס הבין לאומית שהוענקה למספ
קטן של יועצי מיחשוב בלתי תלויים באירופה.

• ספר הלימוד בקורס "יסודות המקרו" ע"י רן אברהמ